DOCUMENT RESUME

ED 201 858 CE 028 969

TITLE	Competency Based Education Curriculum for Data Processing.
INSTITUTION	Radio Corp. of America, Charleston, W. Va. Education Services.; West Virginia State Vocational Curriculum Lab., Cedar Lakes.
SPONS AGENCY	West Virginia State Dept. of Education, Charleston. Bureau of Vocational, Technical, and Adult Education.
PUB DATE	Mar 81
NOTE	160p.
AVAILABLE FROM	West Virginia Vocational Curriculum Laboratory, Cedar Lakes Conference Center, Ripley, WV 25271.
EDRS PRICE	MF01/PC07 Plus Postage.
DESCRIPTORS	Business Education; Check Lists; *Competency Based Education; *Computer Storage Devices; Curriculum Guides; *Data Processing; Data Processing Occupations; *Job Skills; Learning Activities; Learning Modules; Postsecondary Education; Problem Sets; *Programing; *Programing Languages; Secondary Education; Task Analysis
IDENTIFIERS	BASIC Programing Language; COBOL Programing Language; FORTRAN Programing Language

ABSTRACT

This curriculum for data processing is organized into four learning modules. Each module is comprised of four to seven competencies. A student competency sheet provided for each competency is organized into this format: module and competency number and name, performance guide, learning activities, and an evaluation. Where appropriate, student quizzes, work sheets, and information sheets are included. Titles of the modules include (1) Data Processing Fundamentals, (2) BASIC Programming, (3) COBOL Programming, and (4) FORTRAN Programming. (YLB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED201858

Competency Based Education Curriculum
for
DATA PROCESSING

Prepared by

RCA Education Services
A Division of RCA Corporation
Charleston, West Virginia 25302

Produced at

The West Virginia Vocational Curriculum Laboratory
Cedar Lakes Conference Center
Ripley, West Virginia 25271

by

Fred W. Harrington
Curriculum Coordinator

Larry Froehlich
Curriculum Specialist

Macel Rhodes
Secretary

Tom Thurman
Production Manager

Jeanne Ellison
Librarian

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. Thomas

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

This curriculum was developed through the cooperation and support of the Bureau of Vocational, Technical and Adult Education, West Virginia Department of Education, Charleston, West Virginia.

March 1981

CE 028 969

WEST VIRGINIA BOARD OF EDUCATION

EXECUTIVE OFFICES

Charleston, West Virginia 25305

Mr. William M. Smith, President
Follansbee, 1982

Dr. Charles H. Wagoner, Vice President
Weston, 1984

Mr. Paul J. Morris, Secretary
Dunbar, 1980

Mr. James P. Clark, Member
St. Albans, 1986

Mr. Duke A. McDaniel, Member
Petersburg, 1987

Mrs. Mary Martha Merritt, Member
Beckley, 1983

Mr. Bob E. Myers, Member
Huntington, 1985

Dr. Carl J. Roncaglione, Member
Charleston, 1981

Mr. Richard N. Schnacke, Member
New Martinsville, 1988

EX OFFICIO MEMBER
Dr. Roy Truby
Superintendent
West Virginia Department of Education

EX OFFICIO MEMBER
Dr. Robert Ramsey
Chancellor
West Virginia Board of Regents

VOCATIONAL EDUCATION PROGRAMS ARE AVAILABLE WITHOUT REGARD TO RACE, COLOR, NATIONAL ORIGIN, SEX OR HANDICAP.

PREFACE

It is with great pleasure that we bring you another West Virginia competency based education (CBE) curriculum. This competency based Data Processing curriculum was developed under contract for West Virginia. The effort was coordinated by staff of the West Virginia Department of Education, Bureau of Vocational, Technical and Adult Education.

This document is organized into four learning modules. The modules and the competencies for each are given in the program organization. Each competency includes a performance guide, learning activities, and an evaluation.

The competency based Data Processing curriculum was disseminated to West Virginia business and office instructors at six regional inservices conducted in March and April, 1981. Additional copies of this curriculum are available from the:

Vocational Curriculum Laboratory
Cedar Lakes Conference Center
Ripley, West Virginia 25271
372-8673 or toll-free 1-800-642-2670

I trust this material will be of assistance in your program. Any suggestions you may have for improving this curriculum may be sent to our state Curriculum Lab. Through this feedback we plan to continually update and revise our materials.

VOCATIONAL, TECHNICAL AND ADULT EDUCATION

Clarence E. Burdette

Clarence E. Burdette
Assistant State Superintendent of Schools

TABLE OF CONTENTS

PAGE

I. Introduction 3

II. Identification and Validation of Job Titles and Competencies . . 4

III. Functions of the Developmental Committee 5

IV. Program Organization 6

V. Bibliography 7

VI. Implementation of Curriculum 8

INSTRUCTOR'S GUIDE: DATA PROCESSING

I. Introduction

The State Department of Vocational, Technical and Adult Education supports the concept of Competency Based Education because it is based upon occupational relevancy and can account for skills required by an individual to perform on the job. CBE is based on mastery learning principles and requires demonstration of skill and thus can ensure that critical skills required of the entry-level worker are included and attained in training. CBE allows for individual progress, and thus can provide employable skills for some students after short periods of training and can also lead to preparation for advanced jobs.

This instructor's guide is to be used as a reference for instructors and an aid in implementing the curriculum in their program. The following sections are included in the guide.

I. Introduction
II. Identification and Validation of Job Titles and Competencies
III. Functions of the Developmental Committee
IV. Program Organization
V. Bibliography
VI. Implementation of Curriculum

II. Identification and Validation of Job Titles and Competencies

In order to identify and validate job titles and competencies for Data Processing the following sources were reviewed and/or contacted.

Occupational Outlook Handbook (1978-1979)
Dictionary of Occupational Titles
V-TECS Curriculum Catalog (Data Processing) Georgia, 1976
Curriculum Materials from Skyline
Career Development Center (Dallas, Texas)
Data Processing Course Outlines (Wheeling Park High School)
Curricula outlines, program descriptions and reference materials BV (TAE) ERIC

Relevant job titles were obtained as follows:

Data Typist (203,582-022)
Terminal Operator (203.582-054)
Console Operator (213.362-010)
Keypunch Operator
Tape Librarian (206.387.-010)
Detail Programmer (219.367-026)
Programmer (012.167-066 and 020.162-014)

V-TECS Catalog in Data Processing used Computer Operator (213.382-018) and Supervisor of Operation (213.138-010)

This curriculum covers entry level skills in BASIC, COBOL, FORTRAN Programming.

III. Functions of the Developmental Committee

The Developmental Committee was established to guide the development of the curriculum. The committee consisted of Dr. Fred Harrington, State Curriculum Supervisor, John Siminson of Wheeling Park High School and Lola Hunter of Magnolia High School. The committee met on several occasions for the following purposes.

1. Establish curriculum format and clarify project objectives
2. Review major job titles and competencies.
3. Review survey results.
4. Finalize competency list.
5. Review student materials.
6. Establish procedures for prevention of and removal of sex role stereotyping and sex bias.
7. Review and establish reading levels of the curriculum.

IV. Program Organization

The following is a listing of the modules and competencies included in the curriculum.

MODULE 01: DATA PROCESSING FUNDAMENTALS
COMPETENCY 01: Describe flow of data through computer data processing system.
COMPETENCY 02: Use ANSI standard flowchart symbols to follow and create flowcharts.
COMPETENCY 03: Convert between binary, octal, hexidecimal, and decimal number systems.
COMPETENCY 04: Perform arithemtic operations in binary and octal number systems.

MODULE 02: BASIC PROGRAMMING
COMPETENCY 01: Recognize legal variables used in BASIC programming language.
COMPETENCY 02: Identify types and limitations of numbers and constants used in BASIC.
COMPETENCY 03: Identify BASIC Statements and Symbols.
COMPETENCY 04: Code a BASIC program from a simple flowchart, including a loop.
COMPETENCY 05: Flowchart a problem into BASIC.
COMPETENCY 06: Write and run a BASIC program using commands of LET, PRINT, and END.
COMPETENCY 07: Write and run a BASIC program using READ and DATA statements.
COMPETENCY 08: Write and run a BASIC program using IF-THEN, GOTO, and FOR-TO/NEXT statements.

MODULE 03: COBOL PROGRAMMING
COMPETENCY 01: Use required COBOL entries and correct COBOL syntax rules to achieve a compilation.
COMPETENCY 02: Prepare input, output, and work file descriptions for COBOL program requirements.
COMPETENCY 03: Use diagnostic messages for efficient correction of COBOL syntax errors.
COMPETENCY 04: Perform data handling and manipulations by use of COBOL conditional statements and COBOL arithmetic statements.
COMPETENCY 05: Use the editing features of COBOL to improve the readability of output.
COMPETENCY 06: Manipulate data in table form using COBOL program features.

MODULE 04: FORTRAN Programming
COMPETENCY 01: Code and execute simple programs in FORTRAN.
COMPETENCY 02: Use FORTRAN format statements, control statements, and arithmetic operations.
COMPETENCY 03: Debut FORTRAN programs containing syntax errors.
COMPETENCY 04: Use FORTRAN formats I, E, and F for number specifications.

MODULE 04: FORTRAN PROGRAMMING
COMPETENCY 05: Use various formats in preparing readable output to FORTRAN problems.
COMPETENCY 06: Provide for ease of FORTRAN program maintenance by using various documentation techniques.
COMPETENCY 07: Use the logical statements, the TYPE statements, and format statements of FORTRAN.

V. Bibliography of Resources

The following reference books were sold as sources in the development of the curriculum. A note of reminder here, the State Curriculum Laboratory can help you secure the resources you may need.

Awad, Elias M. *Business Data Processing*, Third Edition, Englewood Cliffs, New Jersey: Prentice Hall Publishing Company, 1971.

Farina, Mario *COBOL Simplified* Englewood Cliffs, New Jersey: Prentice Hall Publishing Company, 1965.

Farina, Mario *Fortran IV Self Taught* Englewood Cliffs, New Jersey: Prentice Hall Publishing Company, 1966.

Farina, Mario *Programming in BASIC, The Time Sharing Language* Englewood Cliffs, New Jersey: Prentice Hall Publishing Company, 1968.

Occupational Outlook Handbook (1978-79 Edition), *Dictionary of Occupational Titles* (4th, ed., 1977), V-TECS curriculum catalog (Data Processing, Georgia, 1976)

Business Data Processing. Diablo Valley College, Concord California, 1959.

Computers and Careers: A Suggested Curriculum for Grades 9-12. Central Texas College, Killeen, Texas, 1973.

VI. Implementation of Curriculum

The following guide was developed to aid in conducting the Data Processing program. Each competency sheet is summarized as to hard/soft ware needed, instructor's responsibilities and a summary and/or answers to the Evaluation. The guide does not include regular instructor activities such as supervising student's practice, or checking student progress.

Each instructor's responsibilities will differ because of the different equipment available to programs throughout the State.

STUDENT COMPETENCY SHEET

MODULE 01:	Data Processing Fundamentals
COMPETENCY 01:	Describe flow of data through computer data processing system.

PERFORMANCE GUIDE:

1. Describe flow of data through various input devices.
2. Describe flow of data through control unit.
3. Describe flow of data through main storage.
4. Describe flow of data through logic unit.
5. Describe flow of data through various output devices.

LEARNING ACTIVITIES

1. Read pages 142-147 and 151-159 in Business Data Processing. Answer questions 1, 2, 4, 7, and 10 at the end of the chapter. Have your answers checked.

2. Read pages 200-202 and 207-210 in Business Data Processing. Answer questions 1, 2, 6, 13, and 18 at the end of the chapter. Have your answers checked.

3. Read pages 247-248, 257-262, and 264-266 in Business Data Processing. Answer questions 1, 4, 5, 13, and 14 at the end of the chapter. Have your answers checked.

4. Read pages 272-274 and 285-286 in Business Data Processing. Answer questions 2, 6, and 9 at the end of the chapter. Have your answers checked.

5. Define these terms:
   a. Point-of-Scale (POS) input system
   b. Data recorder
   c. Disk pack
   d. Input device
   e. Output device
   f. Peripheral equipment
   g. Primary storage
   h. Secondary storage
   i. File memory
   j. Logic unit
   k. Central processing unit

EVALUATION

Given an exercise asking for flow of data through input, control unit, logic unit, main storage, and output of computer data processing, describe the flow of data.

STUDENT COMPETENCY SHEET

MODULE 01: Data Processing Fundamentals

COMPETENCY 02: Use ANSI standard flowcharts symbols to follow and create flowcharts.

PERFORMANCE GUIDE:
1. Read or develop problem statement.
2. Establish pacemeters and/or limitations.
3. Sketch sequence.
4. Establish flowchart using symbols.
5. Check and revise flowchart.

LEARNING ACTIVITIES

1. Review pages 374-386 in Business Data Processing. Identify each flowchart symbol.
2. Write the name for each flowchart symbol shown on page 375 in Business Data Processing. Have your work checked.
3. Modify the flowchart on page 384 in Business Data Processing. Include a conditional inner loop if B ≥ 100 and C ≥ 500. Have your work checked.

EVALUATION

Given a description of a situation in which computer data processing can be used, develop a program flowchart using ANSI standard symbols.

EVALUATION 01/02

Develop a flowchart for the following situation. The steps may be rearranged. Use standard ANSI symbols.

Customer Payment Processing

Open payments; run adding machine tape on payments in batches of 50.

Run adding machine tapes on bill stubs received with payments in same batches of 50.

Compare tapes; if partial payments included, place PP code and partial amount on bill stubs.

If no partial payments and tapes do not agree, cycle payments and bills back for reading.

Send payments out for bank deposit.

Read bill stubs into computer and put onto magnetic tape. Bill stubs to file.

Sort payment tape on computer into ascending sequence by account number.

On computer, match payment tape to old customer master tape and subtract payments. Write out new customer master tape. Old customer master tape to file.

On the computer, use payment tape to print out report of payments received. Payment tape to file.

On computer, use new customer master tape to print out report of amount still owed by customers who have made payments. Also, print report of customers who have not made payment for 60 days. New customer master tape to file. Reports to credit department.

STUDENT COMPETENCY SHEET

MODULE 01: Data Processing Fundamentals

COMPETENCY 03: Convert between binary, octal, hexidecimal, and decimal number systems.

PERFORMANCE GUIDE: 1. Convert between binary, octal, hexidecimal, and decimal number systems.

LEARNING ACTIVITIES

1. Study pages 164-167 in Business Data Processing. Write the binary equivalent of the decimal numbers 1-20. Have your work checked.
2. Study pages 168-169 in Business Data Processing. Write the octal equivalent of the decimal numbers 1-32. Have your work checked.
3. Study pages 170-171 in Business Data Processing. Write the hexidecimal equivalent of the decimal numbers 1-96. Have your work checked.
4. Complete problems 9, 12, 13, and 14 on pages 174-175 in Business Data Processing. Have your work checked.

EVALUATION

Given fifteen numbers written binary, octal, and hexidecimal, convert them to the decimal equivalent within 95% accuracy.

EVALUATION 01/03

1. Convert the following numbers to decimal numbers:

   a. 111_2
   b. 1010_2
   c. 10_2
   d. 111110_2
   e. 1010_2

2. Convert the following numbers to decimal numbers:
   a. 32_8
   b. 119_8
   c. 471_8
   d. 46_8
   e. 1112_8

3. Convert the following numbers to decimal numbers:
   a. ooA_{16}
   b. $\mathrm{o74}_{16}$
   c. oBF_{16}
   d. ooF_{16}
   e. $\mathrm{1o3}_{16}$

EVALUATION 01/03

ANSWER KEY

1. a. 7
   b. 21
   c. 2
   d. 62
   e. 10

2. a. 26
   b. 81
   c. 313
   d. 38
   e. 586

3. a. 10
   b. 116
   c. 191
   d. 15
   3. 259

STUDENT COMPETENCY SHEET

MODULE 01:	Data Processing Fundamentals
COMPETENCY 04:	Perform arithmetic operations in binary and octal number systems.
PERFORMANCE GUIDE:	1. Convert decimal numbers to binary and octal numbers. 2. Add, subtract, multiply, and divide binary and octal numbers. 3. Check answers by converting to decimal equivalents.

LEARNING ACTIVITIES

1. Study pages 178-189 in *Business Data Processing*. Work the sample problems and have your work checked.

2. Complete problems 2, 3, 4, 5, and 6 on pages 196-197 in *Business Data Processing*. Have your work checked.

3. Study pages 189-192 in *Business Data Processing*. Work the sample problems and have your work checked.

4. Complete problems 7, 8, 9, 10 and 11 on pages 197-198 in *Business Data Processing*. Have your work checked.

EVALUATION

Given 32 problems involving binary and octal addition, subtraction, multiplication, and division, and any available references, solve the problems with 90% accuracy.

EVALUATION 01/04

Add the following:

1. $\begin{array}{r} 1011_2 \\ +1001_2 \\ \hline \end{array}$

2. $\begin{array}{r} 11111_2 \\ +10001_2 \\ \hline \end{array}$

3. $\begin{array}{r} 10101_2 \\ +\quad 10_2 \\ \hline \end{array}$

4. $\begin{array}{r} 110110_2 \\ +\quad 111_2 \\ \hline \end{array}$

5. $\begin{array}{r} 103_8 \\ +210_8 \\ \hline \end{array}$

6. $\begin{array}{r} 127_8 \\ +200_8 \\ \hline \end{array}$

7. $\begin{array}{r} 1643_8 \\ +\ 107_8 \\ \hline \end{array}$

8. $\begin{array}{r} 512_8 \\ +1117_8 \\ \hline \end{array}$

Subtract the following:

9. $\begin{array}{r} 1011_2 \\ +111_2 \\ \hline \end{array}$

10. $\begin{array}{r} 1111_2 \\ -\quad 1_2 \\ \hline \end{array}$

11. $\begin{array}{r} 10101_2 \\ -1111_2 \\ \hline \end{array}$

12. $\begin{array}{r} 110111_2 \\ -\ 11110_2 \\ \hline \end{array}$

13. $\begin{array}{r} 107_8 \\ -67_8 \\ \hline \end{array}$

14. $\begin{array}{r} 116_8 \\ -77_8 \\ \hline \end{array}$

15. $\begin{array}{r} 1000_8 \\ -217_8 \\ \hline \end{array}$

16. $\begin{array}{r} 1436_8 \\ -1435_8 \\ \hline \end{array}$

Multiply the following:

17. $\begin{array}{r} 111_2 \\ \times 101_2 \\ \hline \end{array}$

18. $\begin{array}{r} 11010_2 \\ \times 10_2 \\ \hline \end{array}$

19. $\begin{array}{r} 10100_2 \\ \times 110_2 \\ \hline \end{array}$

20. $\begin{array}{r} 101111_2 \\ \times 111_2 \\ \hline \end{array}$

21. $\begin{array}{r} 37_8 \\ \times 21_8 \\ \hline \end{array}$

22. $\begin{array}{r} 100_8 \\ \times 16_8 \\ \hline \end{array}$

23. $\begin{array}{r} 156_8 \\ \times 32_8 \\ \hline \end{array}$

24. $\begin{array}{r} 742_8 \\ \times 11_8 \\ \hline \end{array}$

Divide the following:

25. $10_2 \overline{)100_2}$

26. $111_2 \overline{)11011_2}$

27. $1001_2 \overline{)11111_2}$

28. $1_2 \overline{)10010_2}$

29. $7_8 \overline{)52_8}$

30. $6_8 \overline{)1476_8}$

31. $14_8 \overline{)712_8}$

32. $10_8 \overline{)1472_8}$

EVALUATION 01/04

ANSWER KEY

1. 10100_2
2. 110000_2
3. 10111_2
4. 111101_2
5. 313_8
6. 327_8
7. 1752_8
8. 1631_8
9. 100_2
10. 1110_2
11. 110_2
12. 11001_2
13. 20_8
14. 17_8
15. 561_8
16. 1_8
17. 100011_2
18. 110100_2
19. 1111000_2
20. 101001001_2
21. 517_8
22. 1600_8
23. 5624_8
24. 10050_8
25. 10_2
26. 100_2
27. 11_2
28. 10010_2
29. 6_8
30. 96.33_8
31. 31.6_8
32. 67.25_8

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 01: Recognize legal variables used in BASIC programming language.

PERFORMANCE GUIDE:

1. Review BASIC programs that have been coded and identify whether BASIC variables are legal or invalid (illegal).
2. Change any invalid (illegal) BASIC variables to legal variables.
3. Submit program for run.

LEARNING ACTIVITIES

1. Read Lessons 1-7 in Programming in BASIC, The Time Sharing Language. These are short lessons. Complete the exercises for each lesson and have your answers checked.
2. Complete Handout 02/01/02. Have your answers checked.

EVALUATION

Given a list of legal and invalid (illegal) BASIC variables, identify the legal variables with 90% accuracy.

HANDOUT 02/01/02

CIRCLE the legal BASIC Variables on the following list.

1. ALPHA
2. b3
3. i
4. C7)
5. NC5
6. P2
7. x9
8. PRIND
9. A
10. N(5)
11. B
12. I
13. M5
14. PQ
15. X9
16. E
17. A(3,5)
18. B3
19.
20. 6X
21. A
22. CI
23. E
24. H
25. !3
26. 5
27. 7
28. T1
29. J
30. GEE
31. F
32. I
33. Z99
34. Z6
35. 8
36. A9
37. 4
38. KC
39. D1
40. B1

HANDOUT 02/01/02

ANSWERS

6

9

10

11

12

13

15

16

17

18

21

23

24

28

29

31

32

34

36

39

40

EVALUATION 02/01

Write YES or NO to tell whether each of the following is a valid BASIC variable.

1. R ______
2. S ______
3. G0 ______
4. GO ______
5. H47 ______
6. PROD ______
7. 6h ______
8. h6 ______
9. F4 ______
10. AI ______
11. 6H ______
12. (z) ______
13. B6 ______
14. A1 ______
15. B+5 ______
16. I ______
17. (R,1) ______
18. R,7 ______
19. R5 ______
20. i ______

ANSWERS

EVALUATION 02/01

Write YES or NO to tell whether each of the following is a valid BASIC variable.

1. Yes
2. Yes
3. No
4. Yes
5. No
6. No
7. No
8. No
9. Yes
10. No
11. No
12. No
13. Yes
14. Yes
15. No
16. Yes
17. No
18. No
19. Yes
20. No

NOTE: Some versions of BASIC allow variables to exceed the BASIC form (letter, letter + digit, letter + $ or letter + digit + $) in length merely ingoring the excess letters and or numbers.

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 02: Identify types and limitations of numbers and constants used in BASIC.

PERFORMANCE GUIDE:

1. Review BASIC programs that have been coded and identify whether numbers and constants have been correctly used.
2. Change any incorrectly used numbers and constants.
3. Submit program for run.

LEARNING ACTIVITIES

1. Read lessons 9-10 in Programming in BASIC, The Time-Sharing Language. These are short lessons. Complete the exercises for each lesson and have your answers checked.

2. Complete Handout 02/02/02. Have your answers checked.

EVALUATION

Given a list of legal and invalid (illegal) numbers and constants, identify their status, type, and limit as used in BASIC with 90% accuracy.

HANDOUT 02/02/02

Constants, unlike variables, are fixed quantities that do not change during the running of a program. An example of a constant from mathematics is the number PI, which is equivalent to 3.14159··· . In BASIC we have a few different types of constants. It is essential you know all types because the computer can return any one in the form of output at any time. Also, it may be more convenient for you to use a particular type when you are writing a program.

We shall look at three types of constants.

1. decimal format - the number contains a decimal point.
   57.3 -8.37123 .056 -1.28

2. integer format - the number does not contain a decimal point.
   85 200 560045 5

Most BASIC compilers will print decimal and integer constants to a maximum of six digits. For example, if you input the number 12345678, the computer would return the number 1.23457E7. Note the computer rounds to six significant figures. This should not seem so strange if you remember scientific notation, 1.23457 x 10^7.

This number with E in it is in:

3. exponetial format - (also called E format) the number contains the letter E, with the mantissa appearing before the E and the exponent appearing after the E.

   mantissa E exponent

   1.23457E7

5

The number 5E-8 would be .00000005 in decimal form. You can see exponential format is a very effective shorthand for writing very large and small numbers.

Here are some examples of constants in BASIC.

Constant	Type
87.23	decimal
57	integer
5E4	exponetial

BASIC will not accept the following illegal constants.

123,456	comma not allowed
5.39E86	number too large

Converting to and from exponential form to ordinary form is done as follows:

1. exponential to ordinary
   8.23523E6 = ?

   Since the "6" means multiply by 10^6, move the decimal point 6 places to the right.

   8.23523E6 = 8.23523 = 8235230.
   6 places

   Of course if the exponent is negative, move the decimal point to the left.

2. ordinary to exponential

   11111111 = 1.11111E7

   .000087251 = 8.7251E-5

Complete the following exercises.

I. For each constant indicate which type (decimal, integer, exponential) it is. However, if the constant is illegal indicate this and tell why.

1. 654321 ______________
2. 5E4 ______________
3. 6.875,64 ______________
4. 3.1416 ______________
5. -823 ______________
6. 2.3E 8 ______________
7. 0 ______________
8. E15 ______________
9. 1.756E ______________
10. 1E-23 ______________

II. Converting form exponential form.

1. 6.275E3 ______________
2. 8.7351E-2 ______________
3. 1.23456E11 ______________
4. 7.265E5 ______________
5. 1E-7 ______________

III. Converting to exponential form.

1. 100000000 ________________

2. 123456789 ________________

3. .027351 ________________

4. 1890. ________________

5. 5.37×10^{14} ________________

HANDOUT 02/02/02

ANSWERS

I.

1. Integer
2. Exponential
3. Decimal (Illegal - commas are not allowed)
4. Decimal
5. Integer
6. Exponent (Illegal - spaces are not allowed in the exponent)
7. Integer
8. Exponent (Illegal - mantissa must be present)
9. Exponent (Illegal - exponent must be present)
10. Exponent

II.

1. 6275.
2. .087351
3. 1.23456000000.
4. 7.26500.
5. .0000001

III.

1. 1E8
2. 1.23456789E8
3. 2.7351E-2
4. 1.890E3
5. 5.37E14

EVALUATION 02/02

Complete the blank columns for each of the following terms.

	Term	Status	Type	Limit
Example 1:	999.999	legal	constant (real)	11 digits
Example 2:	41.0E69	illegal	Exponent	4.314E68

1. 1
2. 123456
3. 12345678901
4. 1.234E11
5. 1E9
6. 1.E-09
7. .69
8. 3.6356295141
9. -456
10. +1.6E42
11. -.267E-40
12. -01.E-50
13. E35
14. .3E**.5
15. .33333E-10
16. .000,34
17. 1,100,000.00E20
18. 1.732415623.E-2
19. 21.146
20. 1E2

ANSWERS

EVALUATION 02/02

1.	Legal	Constant	11 digits
2.	Legal	Constant	11 digits
3.	Legal	Constant	11 digits
4.	Legal	Exponent	4.314E68
5.	Legal	Exponent	4.314E68
6.	Legal	Exponent	4.314-68
7.	Legal	Constant	11 digits
8.	Legal	Constant	11 digits
9.	Legal	Constant	11 digits
10.	Legal	Exponent	4.314E68
11.	Legal	Exponent	4.314E-68
12.	Legal	Exponent	4.314E-68
13.	Illegal	Exponent	4.314E68
14.	Illegal	Exponent	4.314E68
15.	Legal	Exponent	4.314E-68
16.	Illegal	Constant	11 digits
17.	Illegal	Exponent	4.314E68
18.	Illegal	Exponent	4.314E-68
19.	Legal	Constant	11 digits
20.	Legal	Exponent	4.314E68

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 03: Identify BASIC Statements and Symbols.

PERFORMANCE GUIDE:

1. Review BASIC programs that have been coded and identify statement and symbol syntax errors.
2. Change coding to correct statement and symbol syntax errors.
3. Submit programs for run.

LEARNING ACTIVITIES

1. Read lessons 11-14 in Programming in BASIC, The Time-Sharing Language. These are short lessons. Complete the exercises for each lesson and have your answers checked.

2. Get your manufacturer's BASIC reference manual and review the sections on statement and symbol syntax errors.

3. Complete Handout 02/03/03. Have your answers checked.

EVALUATION

Given a list of BASIC statements and symbols, identify invalid (illegal) items and correct syntax errors with 90% accuracy.

HANDOUT 02/03/03

In order to speak a language you must know certain words. In BASIC we will see how certain keywords form the basis for BASIC statements provided correct syntax is used. In other words, you must "speak" to the computer exactly the way the computer will understand you. Not the slightest bug (error) will be accepted. When you learn some statements from BASIC you will be able to put them together to write a BASIC program.

The following is a listing of the most used BASIC statements. We will learn others later when needed. Remember you are interested in the correct syntax (exactly how each is written) now. Pay very close attention to detail. Later, when the various statements are used in programs you will understand fully how the statements are used.

DATA - used with a READ statement to store values

```
 20 READ R,S
  .
  .
  .
150 DATA 25,13.1
```

END - tells the computer the program is ended. Must be the highest numbered statement in program.

```
300 END
```

FOR - TO - first statement in a FOR-NEXT loop tells the computer how many times to execute a loop.

```
 70 FOR I=1 TO 8
  .
  .
  .
120 NEXT I
```

if increment is other than one use STEP

```
130 FOR J=1 TO 20 STEP 2
  .
  .
  .
190 NEXT J
```

J takes on values of 1,3,5,...,17,19

GO TO - unconditional control statement program branches to listed line number

```
 80 GO TO 150
  .
  .
  .
150 ........
```

IF - THEN - - conditional control statement

program branches to listed line number
if condition is true

otherwise, branches to next statement

```
210 IF B=3 THEN 300

220 .......
 .
 .
 .
300 .......
```

INPUT - used to store values directly from the keyboard

```
100 INPUT X
```

when program is run a "?" will appear on the
terminal as a signal to enter data

LET - this assignment statement is used to store values
directly from the program the expression to the
right is calculated and the result is stored under
the variable to the left of the equal sign

```
55 LET X=A*B
```

NEXT - always the last statement in a FOR-NEXT loop
see FOR - TO statement for example

PRINT - outputs the values of variables listed
outputs messages between quotes
commas and semicolons are used for spacing

```
15 PRINT L, W, A
 .
 .
 .
280 PRINT "THERE WERE" N "AREAS CALCULATED."
```

REM - short for remark
non-executable statement used for information

```
10 REM    TRIANGLE PROGRAM
```

STOP - used to terminate execution of a program at
places other than END statement

```
255 STOP
 .
 .
 .
```

READ - used with DATA statement to store values
see DATA for example

Complete the following exercises.

I. Write YES or NO to tell whether each of the following is a valid BASIC statement.

________ 1. 50 STOP AT 200

________ 2. 20 READ C,A,T

________ 3. 100 PRINT "AVERAGE= " A, "TIME = "T

________ 4. 70 REM JOHN JONES PROBLEM #5

________ 5. 60 READ (X,Y)

________ 6. 130 LET A*B=C

________ 7. 150 LET X=Y/Z

________ 8. 200 DATA 5,7,3/5,8,5.1;9

________ 9. 210 DATA -15,-1,0,7,9

________ 10. 20 LET I=.05

________ 11. 90 IF R S THEN 120

________ 12. 180 GO TO "210"

________ 13. 230 FOR X=10 TO 1 STEP -1

________ 14. 110 INPUT A?

________ 15. 97 END

II. Correct any errors in these BASIC statements. Each has an error.

1. 10 IF R=5, THEN 30 ____________________

2. 20 LET X=2Y-3 ____________________

3. 30 NEXTH ____________________

4. 40 INPUT A;B,X ____________________

5. 50 GO TO 50 ____________________

6. 60 READ X,Y/Z ____________________

7. 70 PRINT "ANSWER IS" T (SECONDS) ____________________

8. 80 DATA 13,1/2,8.2 ____________________

9. 90 FOR I=100 TO 1 ____________________

10. 95 END. ____________________

ANSWERS TO EXERCISES

I.

1. No
2. Yes
3. No
4. Yes
5. No
6. No
7. Yes
8. No
9. Yes
10. Yes
11. No
12. No
13. Yes
14. No
15. Yes

II.

1. `10 IF R=5 THEN 30`
2. `20 Let X=2*Y-3`
3. `30 NEXT H`
4. `40 INPUT A,B,X`
5. `50 GO TO ? (not so)`
6. `60 READ X,Y,Z`
7. `70 PRINT 'ANSWER IS' T, '(SECONDS)'`
8. `80 DATA 13,1.5,8.2`
9. `90 FOR I=100 TO 1 STEP-1`
10. `95 END`

EVALUATION 02/03

PART I

Identify the legal BASIC statements by placing an "L" in the space. Identify the illegal statements by placing an "I" in the space. Remember, the statement will be keypunched as it is.

_______ a. 100 DATA 5, 4, 3, 0, 0
_______ b. 200 READ X, Y, A, B, Z
_______ c. 0300 RESTOER
_______ d. PRINT "R="P, "R="R
_______ e. 0500 LET A=X=O
_______ f. 0600 ON 1 GO TO 700,600
_______ g. 0700 IF A/NE O GO TO 600
_______ h. 0800 DIM A(32,22,32)
_______ i. 900 REM A MESS
_______ j. 1010 STOP
_______ k. 1100 LET A=B=C=D
_______ l. 1200 INPUT D
_______ m. 1300 FOR Q=1000 TO 10 * 1000
_______ n. 1400 GO TO 200
_______ o. 1500 FOR A=1 TO 100 STEP 2
_______ p. 1600 STOP
_______ q. 1700 LET R=.05
_______ r. 1800 NEXT Q
_______ s. 1900 NEXTA
_______ t. 2000 END

Part 2

Correct the syntax errors in the following lines of coding. Leave the statements which do not contain syntax errors blank.

a. 10 LET X=5
b. 20 READ Y,Z
c. 30LER W=Y * Z
d. 40 PRINT X
e. 50 LET A * B=X
f. 60 READ W
g. 70 GO TO 9010
h. 80 LET S= (AX+B) * (CX+D)
i. 90 LET A1
j. 100 GO BACK TO 10
k. 110 LET X = 3* 4+5)
l. 120 DATA 5, 11, 25, -5
m. 130 PRINT W↑2
n. 140 GO TO 60
o. 150 DATA 2, 67, -1,
p. 160 STOP
q. 170 END

EVALUATION 02/03

ANSWERS

PART I

a. L

b. L

c. I

d. I

*e. I

f. L

g. I

*h. I

i. L

j. L

*k. I

l. L

m. I

n. L

o. L

p. L

q. L

r. L

s. I

t. L

*Some versions of the language may allow this.

PART II

c

e

h

i

j

k

o

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 04: Code a BASIC program from a simple flowchart, including a loop.

PERFORMANCE GUIDE:

1. Code program on coding sheet.
2. Check coding for omissions and inconsistencies.
3. Enter program codes in preparation for compile.
4. Proofread codes entered.
5. Enter job program codes in preparation for compile.
6. Submit program for run.
7. Debug program using diagnostics.

LEARNING ACTIVITIES

1. Review lessons in Programming in BASIC, The Time-Sharing Language on INPUT, READ, IF, LET, FOR/NEXT, PRINT, STOP, and END statements.

2. Review comparable sections in your manufacturer's BASIC reference manual.

3. Prepare and compile programs for the two problems in Handout 02/04/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, manufacturer's BASIC reference manual, data entry equipment, and coding sheets, prepare the required BASIC program and achieve an error-free compile within three computer runs.

HANDOUT 02/04/03

PROBLEM 1. Which would you prefer: A job that pays $1,000 for the first day and increases $1,000 each day or one that has a salary of 1¢ the first day and is doubled each day till the thirtieth day?

Your output should include what is paid each day for both jobs.

Also, print the total amount paid for the thirty days for each job.

PROBLEM 2. Area of a Circle

The formula for the area of a circle is

A = r , where = 3.14159...
r=radius of a circle

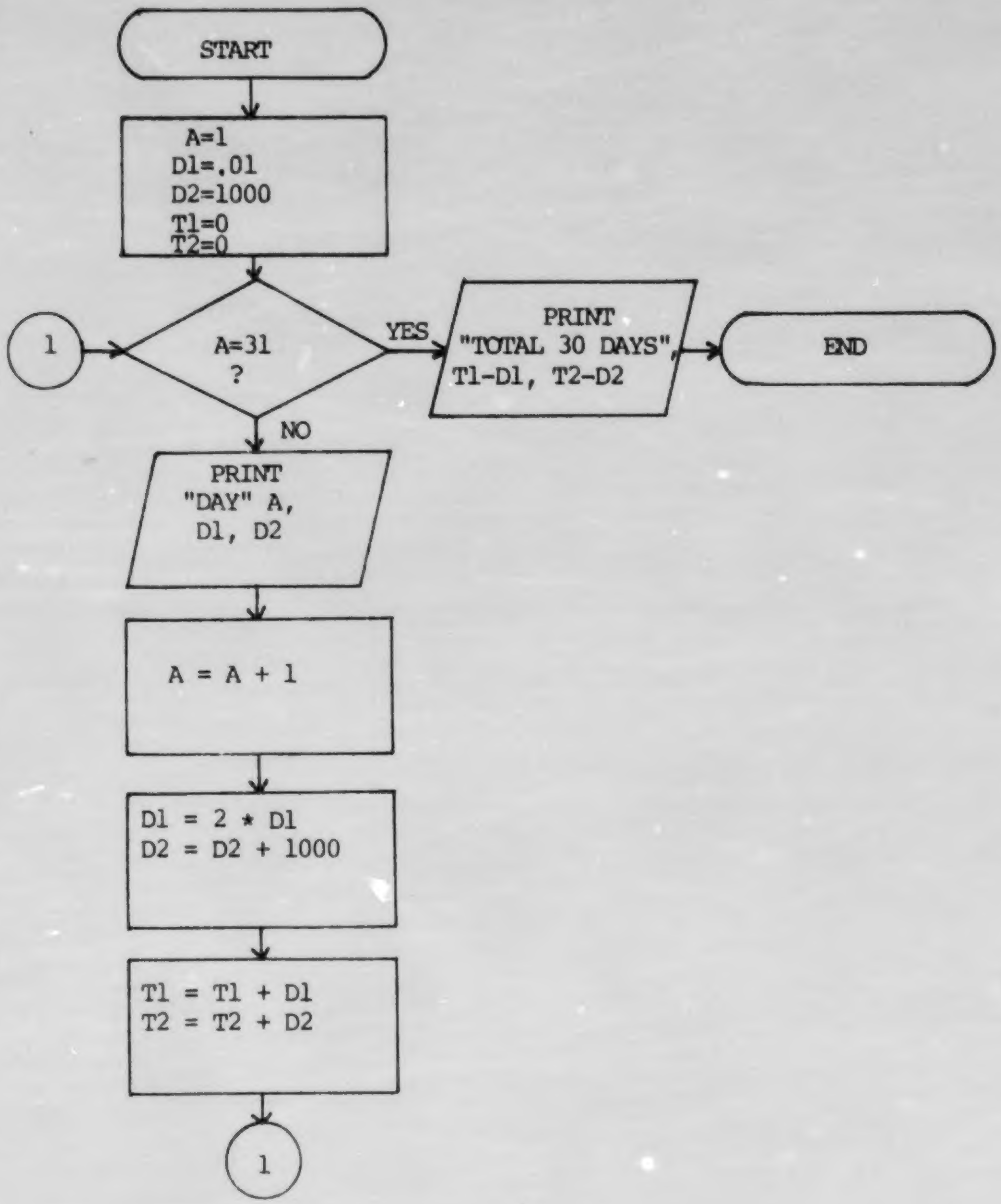
Since (we'll write it as PI) is an irrational number, we can only approximate it on the computer.

You will find the area of a number of circles given the radius of each. Using 3.14, 22/7, 3.1416, and 3.14159 for PI, calculate the area of each circle four times with the values listed for PI.

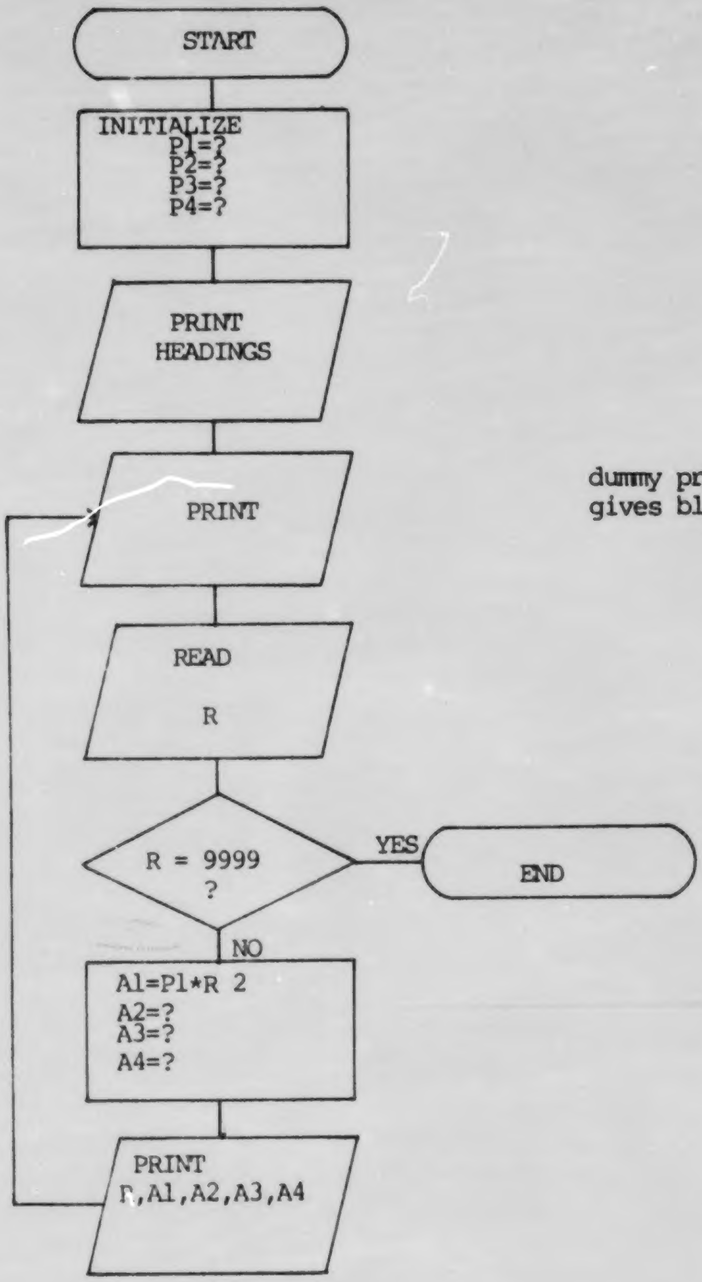
Use READ and DATA statements to enter values for the radii with the last radius to be 9999 to indicate end of data. Print headings for each of the five columns as shown.

```
        AREA OF CIRCLES - DIFFERENT VALUES OF PI

     RADIUS      3.14      22/7    3.1416      3.14159
        .          .         .        .           .
        .          .         .        .           .
        .          .         .        .           .
        .          .         .        .           .
```

FLOWCHART 02/04/03


START
A=1
D1=.01
D2=1000
T1=0
T2=0
1
A=31
?
YES
PRINT
"TOTAL 30 DAYS",
T1-D1, T2-D2
END
NO
PRINT
"DAY" A,
D1, D2
A = A + 1
D1 = 2 * D1
D2 = D2 + 1000
T1 = T1 + D1
T2 = T2 + D2
1

FLOWCHART 2   02/04/03


START
INITIALIZE
P1=?
P2=?
P3=?
P4=?
PRINT
HEADINGS
PRINT
READ
R
R = 9999
?
YES
END
NO
A1=P1*R 2
A2=?
A3=?
A4=?
PRINT
R,A1,A2,A3,A4

dummy print statement
gives blank line of output

EVALUATION 02/04

AREA OF A TRIANGLE

The formula for the area of a triangle is

$A=\frac{1}{2}bh$ or A=.5*B*H

where B is the base and H is the height.

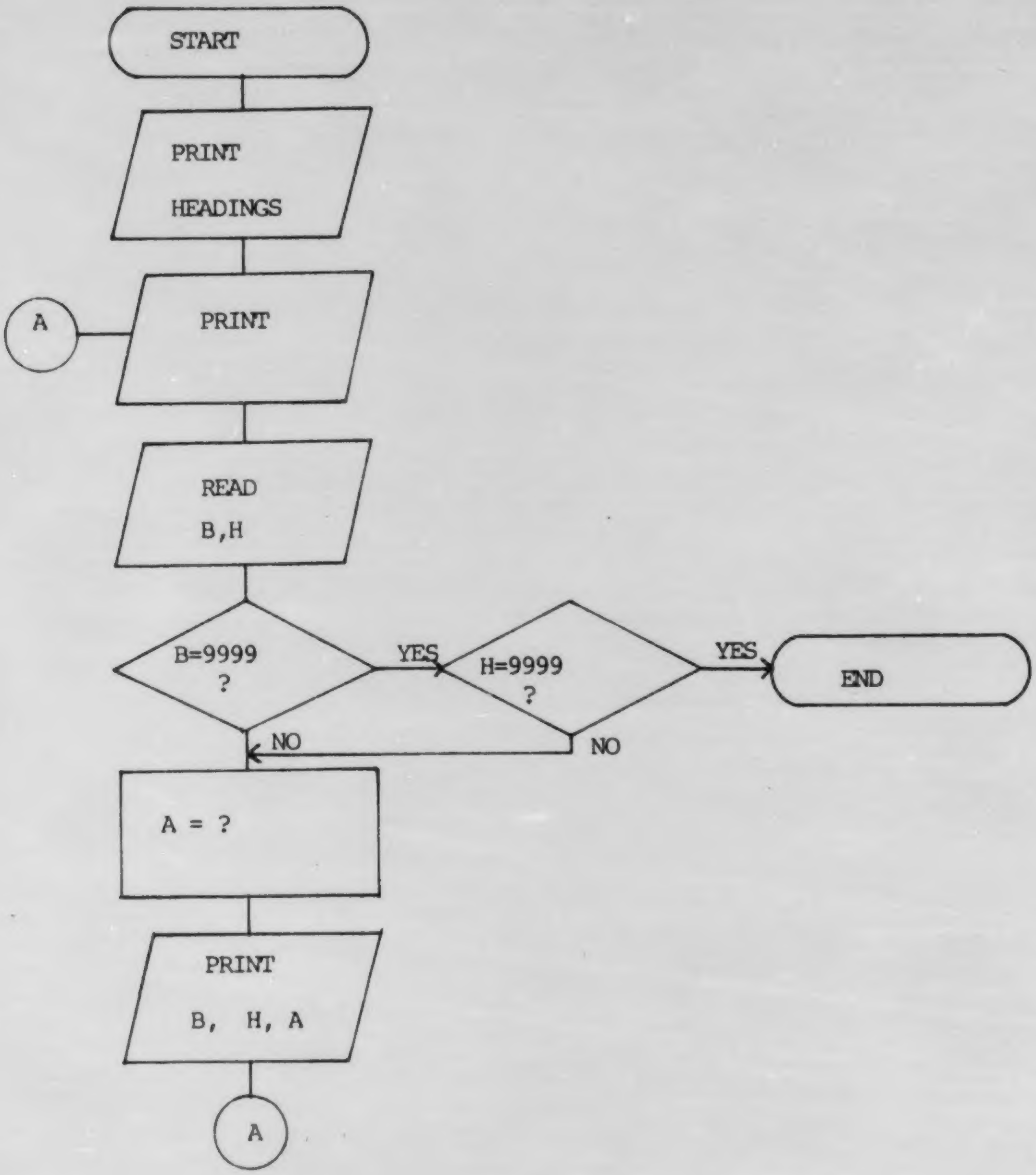
You will have to input two numbers at a time for this problem.

Only when both numbers are 9999 do you end the program.

Output should be a three-column table showing base, height, and area with appropriate column headings.

02/04

FLOWCHART


START
PRINT HEADINGS
A
PRINT
READ B,H
B=9999 ?
YES
H=9999 ?
YES
END
NO
NO
A = ?
PRINT B, H, A
A

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 05: Flowchart a problem into BASIC

PERFORMANCE GUIDE:

1. Review problem statement and input requirements.
2. Review or develop output requirements and layout.
3. Prepare flowchart to accommodate problem statement and output requirements.

LEARNING ACTIVITIES

1. Read lesson 18 in Programming in BASIC, The Time-Sharing Language. Complete the exercises for the lesson and have your answers checked.

2. Complete Handout 02/05/02. Use the procedures listed above in Performance Guide. Have your answers checked.

EVALUATION

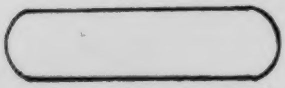
Given problem statement and an ANSI standard flowchart template, prepare an accurate flowchart for the problem statement.

HANDOUT 02/05/02

After gaining experience writing BASIC programs in your last objective, let's put your programming skills to a tougher test. As we said last time, program logic must be completely understood before one can write the program to solve the problem. So now we'll start with the problem and ask you to draw a flowchart.

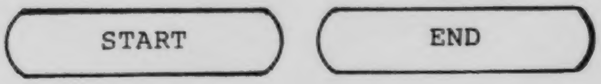
Flowcharting symbols that you have seen so far are basic to program logic. There are many more symbols that only an advanced programmer need be concerned with using.

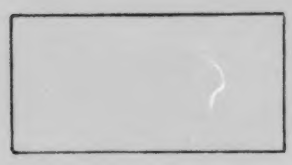
Basic Flowchart symbols



TERMINAL

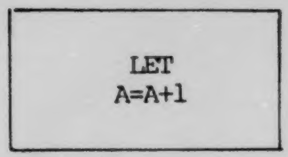
used for starting and stopping a program

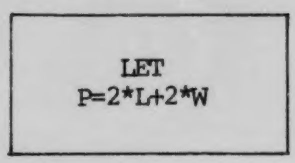


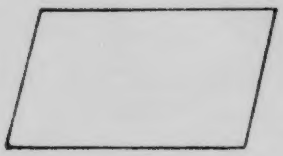


ASSIGNMENT

used to assign values to a variable

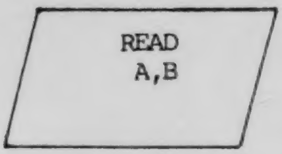


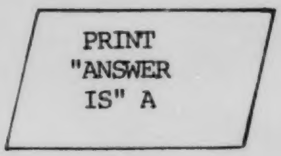


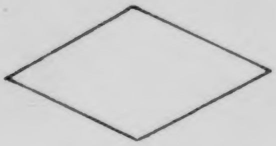


INPUT/OUTPUT

used to store information in the computer or extract it

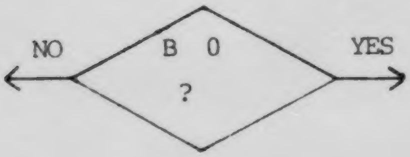


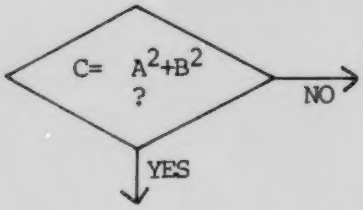

PRINT
"ANSWER
IS" A



DECISION

used when more than one program logic flow can be made




C= A^2+B^2
?
NO
YES

The only way to become a good programmer is to practice preparing flowcharts. "Learn by Doing". So here are some more problems for practice.

Problem 1 - Arithmetic Tables

Write a program to print out arithmetic tables up to 10+10=20, 10-10=0, 10x10=100, 10/10=1 for the four basic arithmetic operations.

Have the computer generate its own data using FOR-NEXT loops.

Output should be by type of operation. To save room do addition and subtraction together and multiplication and division together as shown below. Appropriate headings and spacing should be employed.

```
                    - - - - - -TITLE- - - - - -


     ADDITION TABLES                     SUBTRACTION TABLES

        1 + 1 = 2                            1 - 1 = 0
        1 + 2 = 3                            1 - 2 =-1
            .                                    .
            .                                    .
            .                                    .
        1 + 10=11                            1 -10 =-9
         spaces                               spaces
        2 + 1 = 3                            2 - 1 = 1
        2 + 2 = 4                            2 - 2 = 0
            .                                    .
            .                                    .
            .                                    .
        2 +10 =12                            2 -10 =-8
         spaces                               spaces
            .                                    .
            .                                    .
            .                                    .
            .                                    .
         spaces                               spaces
        10 +1 =11                            10- 1 = 9
            .                                    .
            .                                    .
        10+10 =20                            10-10 = 0
```

```
                             spaces
          MULTIPLICATION TABLES             DIVISION TABLES

                1 x 1 = 1                      1 / 1 = 1
                1 x 2 = 2                      1 / 2 = .5
                    .                              .
                    .                              .
                    .                              .
                1 x10 = 10                     1 /10 = .1
                  spaces                         spaces
                2 x 1 = 2                      2 / 1 = 2
                2 x 2 = 4                      2 / 2 = 1
                    .                              .
                    .                              .
                    .                              .
                2 x10 = 20                     2 /10 = .2
                  spaces                         spaces
                    .                              .
                    .                              .
                    .                              .
                    .                              .
                  spaces                         spaces
               10 x 1 = 10                    10 / 1 = 10
               10 x 2 = 20                    10 / 2 = 5
                    .                              .
                    .                              .
                    .                              .
               10 x10 = 100                   10 /10 = 1
```

Prepare a flowchart to make sure you understand the logic of the problem.

PROBLEM 2 - Calculating Grade Point Average

Statistics gives us three measures of central tendency to help us find the "average" of a set of numbers. The term you have probably always used for "average" is actually the "arithmetic mean" of a set of numbers.

In many schools point values are given for letter grades (A=4, B=3, C=2, D=1, E=O). We'd like to have a program that accepts number equivalents for grades (4,3,2,1,0) for each student. Then for each student print out his/her student number, grades, and grade point average. Different students may have a different number of grades.

When all grades have been entered for all students, print out the class grade point average. Be careful. Class average is not the average of all student averages.

Also, as each student's grade point average is calculated classify it according to the following categories.

4.00
3.99 - 3.00
2.99 - 2.00
1.99 - 1.00
.99 - 0.00

Your output should be similar to the sample on the next page.

```
                TWO BIT  SCHOOL GRADE REPORT

        Student #2375

           Grades are 4,2,4,3,3,4

           G.P.A. is  3.33333

                .
                .
                .
                .
        Student #4328

           Grades are 3,2,4,4,3

           G.P.A.  is 3.2

        THE CLASS GRADE POINT AVERAGE IS    ????

        STUDENT GRADES ARE DISTRIBUTED

              G.P.A.        # of students
               4.00               ??
          3.99 - 3.00             ??
          2.99 - 2.00             ??
          1.99 - 1.00             ??
          0.99 - 0.00             ??
```

EVALUATION 02/05

PROBLEM STATEMENT: Print out a multiplication table up to 12x12=144.

Do not use the multiplication symbol in the program.

INPUT: NONE

OUTPUT: Generate the multiplication tables. Skip five lines between each table as shown in the example below.

EXAMPLE:

```
1 x 1 = 1
1 x 2 = 2
    .
    .
    .
1 x 12 = 12

 (5lines)


2 x 1 = 2
    .
    .
    .
    .
    .
    .
2 x 12 = 24

(5 lines)


12 x 1 = 12
     .
     .
     .
12 x 12 = 144
```

Flowchart - No flowchart is given in this problem. You are to prepare your own flowchart and have your instructor check it before you begin coding.

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 06: Write and run a BASIC program using commands of LET, PRINT, and END.

PERFORMANCE GUIDE:

1. Code program on coding sheets.
2. Check coding for omissions and inconsistencies.
3. Enter program codes in preparation for compile.
4. Proofread codes entered.
5. Enter job program codes in preparation for compile.
6. Submit program for run.
7. Debug program using diagnostics.
8. Test program using test data.

LEARNING ACTIVITIES

1. Review lessons in Programming in BASIC, The Time-Sharing Language, on LET, PRINT, and END commands.

2. Review comparable sections in your manufacturer's BASIC reference manual.

3. Prepare and compile programs for the two problems in Handout 02/06/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's BASIC reference manuals, data entry equipment and coding sheets, prepare the required BASIC program and achieve an error-free compile within four computer runs.

Once a binary number is calculated, the octal equivalent can be gotten by the following relation.

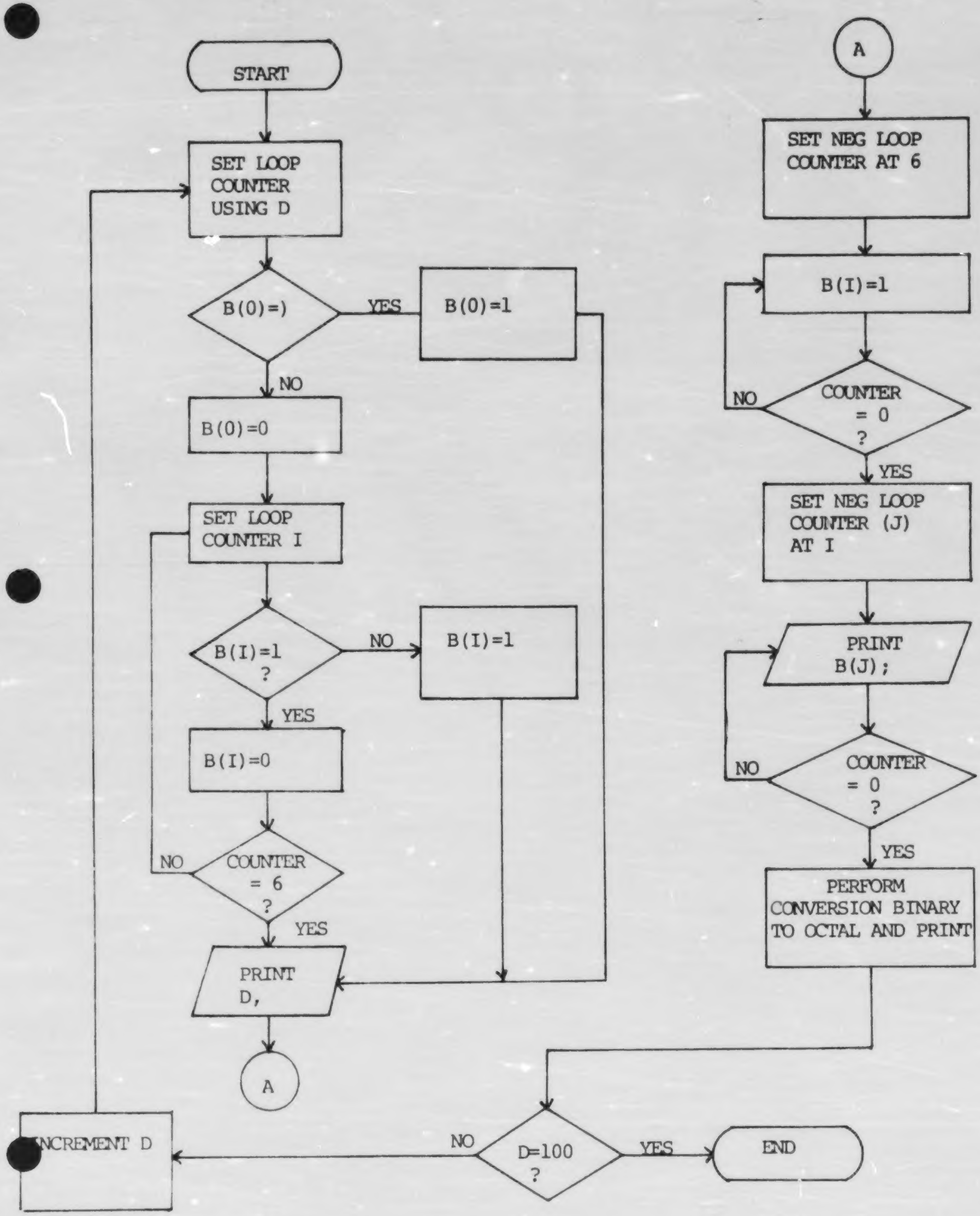
$$63_{10} = \overbrace{111}^{7}\ \overbrace{111}^{7}{}_2 = 77_8$$

$$64_{10} = \overbrace{1}^{1}\ \overbrace{000}^{0}\ \overbrace{000}^{0}{}_2 = 100_8$$

$$100_{10} = \overbrace{1}^{1}\ \overbrace{100}^{4}\ \overbrace{100}^{4}{}_2 = 144_8$$

The flowchart on the next page should be a help in getting started on this problem.

FLOWCHART


START
SET LOOP COUNTER USING D
B(0)=)
YES
B(0)=1
NO
B(0)=0
SET LOOP COUNTER I
B(I)=1 ?
NO
B(I)=1
YES
B(I)=0
NO
COUNTER = 6 ?
YES
PRINT D,
A
INCREMENT D
NO
D=100 ?
YES
END
A
SET NEG LOOP COUNTER AT 6
B(I)=1
NO
COUNTER = 0 ?
YES
SET NEG LOOP COUNTER (J) AT I
PRINT B(J);
NO
COUNTER = 0 ?
YES
PERFORM CONVERSION BINARY TO OCTAL AND PRINT

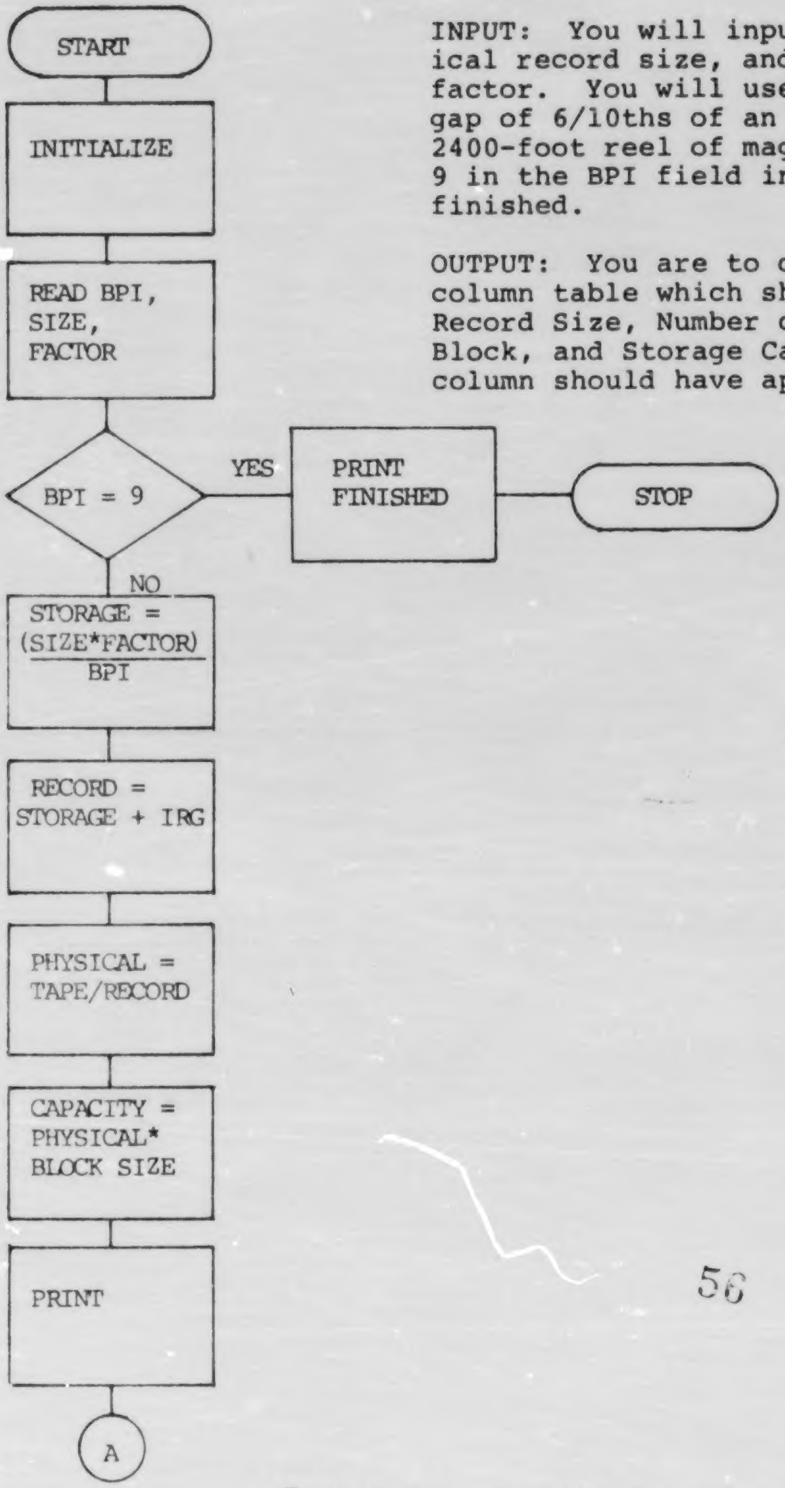
HANDOUT 02/06/03

FLOWCHART

PROBLEM 1: Storage Capacity of Magnetic Tape

INPUT: You will input the BPI, logical record size, and the blocking factor. You will use an inter-record gap of 6/10ths of an inch and a standard 2400-foot reel of magnetic tape. A 9 in the BPI field indicates you are finished.

OUTPUT: You are to output a five-column table which shows the IRG, BPI, Record Size, Number of Records per Block, and Storage Capacity. Each column should have appropriate labels.


START
INITIALIZE
READ BPI, SIZE, FACTOR
BPI = 9
YES
PRINT FINISHED
STOP
NO
STORAGE = (SIZE*FACTOR) / BPI
RECORD = STORAGE + IRG
PHYSICAL = TAPE/RECORD
CAPACITY = PHYSICAL* BLOCK SIZE
PRINT
A

Test Data: Use the data in the examples.

PROBLEM 2. Decimal to Binary Conversion

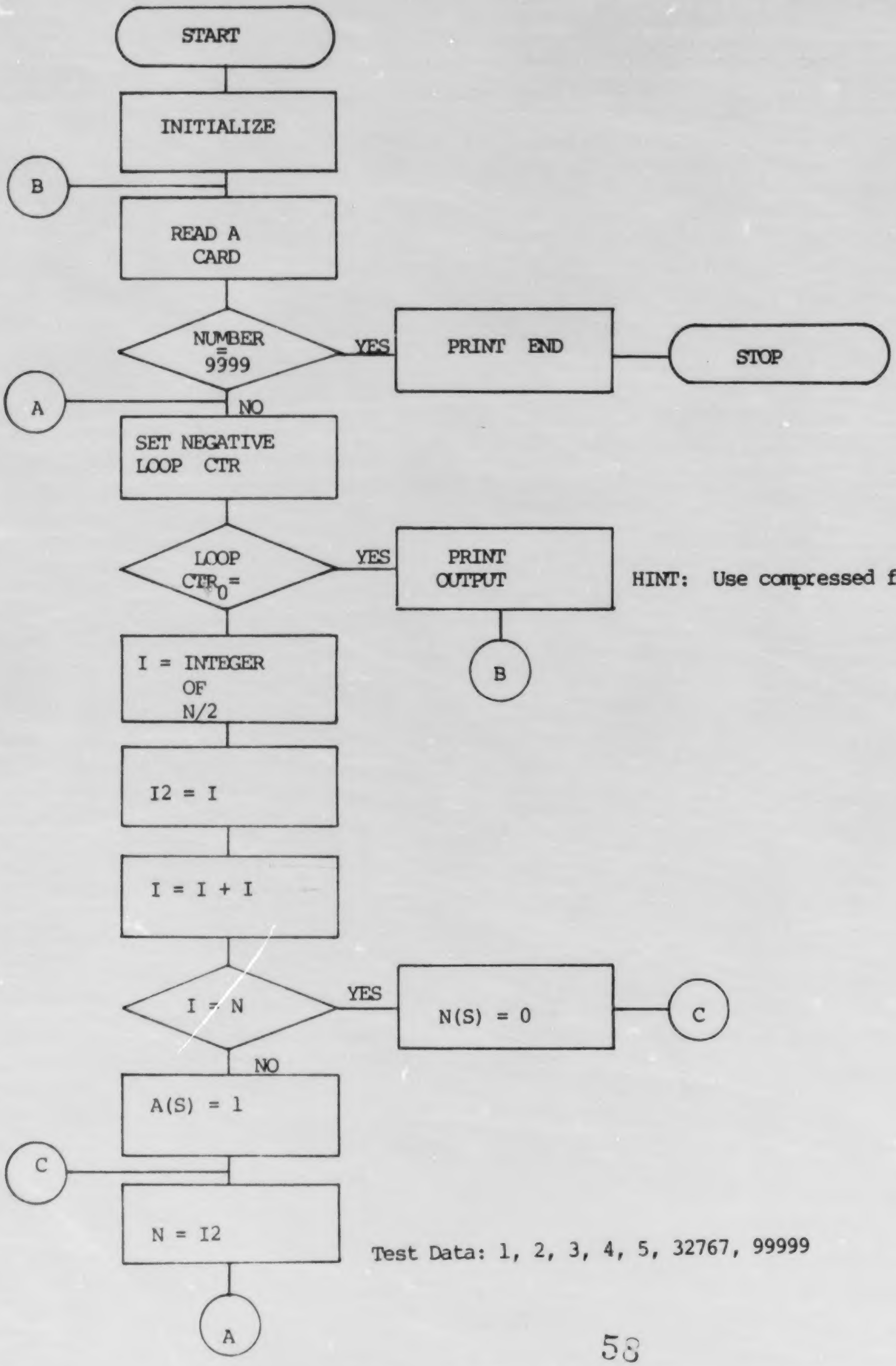
In the Introduction module you studied how to convert decimal numbers to binary numbers. In this problem you are going to write a program to do it. Go review your notes on how to convert decimal to binary. Review the MOD, DIU, and INT functions in the manufacturer's manual on BAS FC.

Input: Your input will be a decimal integer of less than 32768. A field of all 9s indicates you are finished.

Output: Print the following line for each number.

"DECIMAL XXXX EQUALS XXXXXXXXXX in BINARY"

FLOWCHART


START
INITIALIZE
B
READ A CARD
NUMBER = 9999
YES
PRINT END
STOP
A
NO
SET NEGATIVE LOOP CTR
LOOP CTR = 0
YES
PRINT OUTPUT
B
I = INTEGER OF N/2
I2 = I
I = I + I
I = N
YES
N(S) = 0
C
NO
A(S) = 1
C
N = I2
A

HINT: Use compressed format

Test Data: 1, 2, 3, 4, 5, 32767, 99999

EVALUATION 02/06

Binary-Octal-Decimal Equivalents

You are to prepare a table that shows the binary and octal equivalents for decimal integers from 1 to 100.

From your previous study of number systems you should be familiar with base two and its role in computer logic. Base two gives rise easily to base eight and base sixteen which are also used in computers.

The output from your program should be similar to:

```
DECIMAL                BINARY                  OCTAL

      1                     1                      1
      2                    10                      2
      .                     .                      .
      .                     .                      .
      7                   111                      7
      8                  1000                     10
      9                  1001                     11
     10                  1010                     12
      .                     .                      .
      .                     .                      .
     16                 10000                     20
      .                     .                      .
      .                     .                      .
     63                111111                     77
     64               1000000                    100
      .                     .                      .
      .                     .                      .
    100               1100100                    144
```

Since 2^6= 64 and 2^7= 128 and 8^1= 8, 8^2= 64 and 8^3= 512, we will need at most a seven digit number in base two to reach 100 in base ten and a three digit number in base eight to reach 100 in base ten.

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 07: Write and run a BASIC program using READ and DATA statements.

PERFORMANCE GUIDE:

1. Write data descriptions, including subroutines for repetitive functions.
2. Code program on coding sheets.
3. Check coding for omissions and inconsistencies.
4. Enter program codes in preparation for compile.
5. Proofread codes entered.
6. Enter job program codes in preparation for compile.
7. Submit program for run.
8. Debug program using diagnostics.
9. Test program using test data.

LEARNING ACTIVITIES

1. Read lesson 17 in Programming in BASIC, The Time-Sharing Language. Complete the exercises at the end of the lesson and have your answers checked.

2. Review comparable sections on subroutines in your manufacturer's BASIC reference manual.

3. Prepare and compile programs for the two problems in Handout 02/07/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data manufacturer's BASIC reference manual, data entry equipment, and coding sheets, prepare the required BASIC program and achieve an error-free compile within four computer runs.

HANDOUT 02/07/03

PROBLEM 1. Compound Interest

PROBLEM STATEMENT: You want to get a loan from your local bank but you don't know exactly how much the total repayment amount will be; nor how much effect the number of years, interest rates, or frequency of compounding will have on the total repayment amount. You are going to write a program which will calculate the total repayment amount based on the variables. By doing this you can change the variables until you find the best amount.

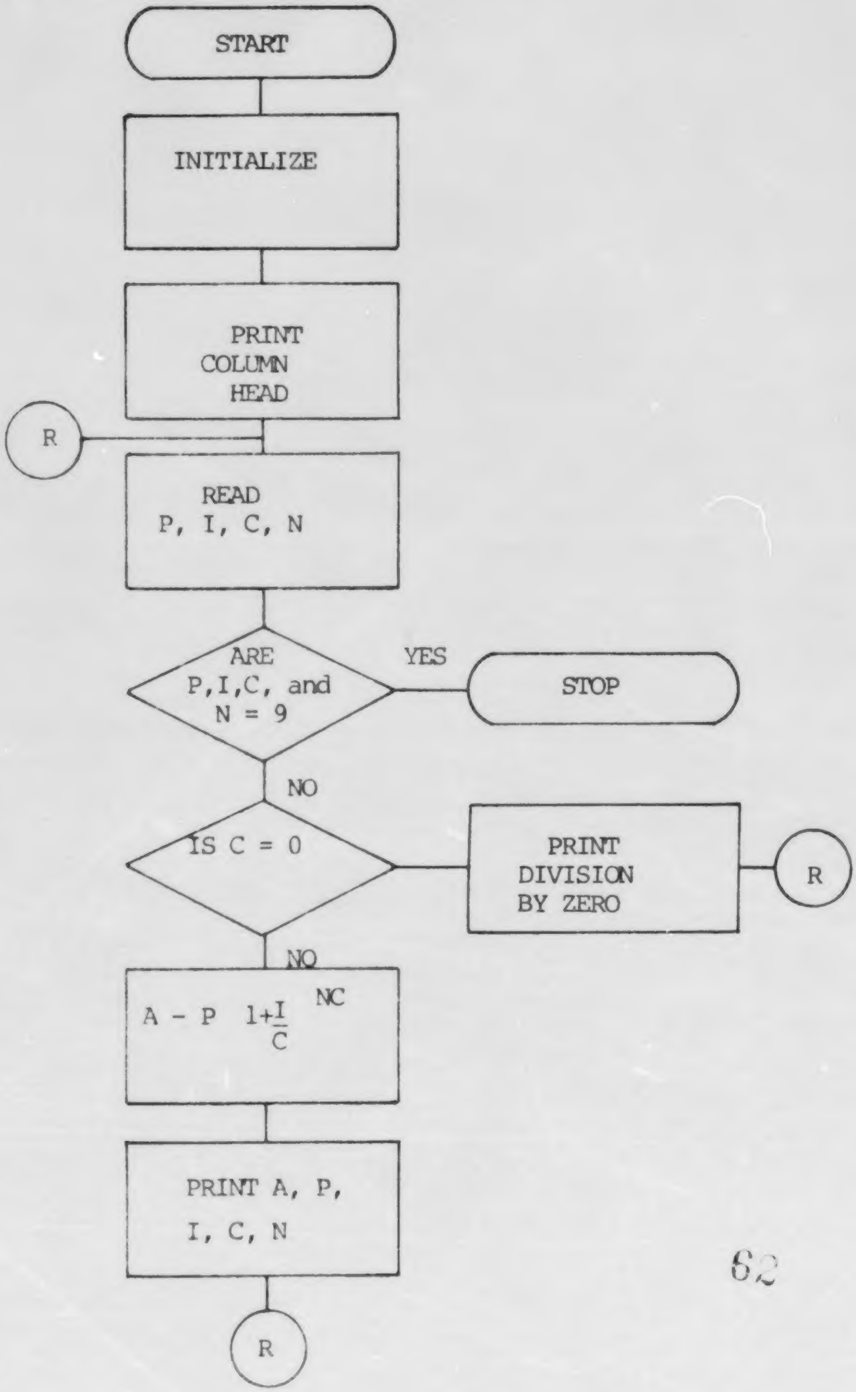
The formula for compounding interest is: $A = P\left(1 + \frac{I}{C}\right)^{NC}$ where

A is the amount (principal + interest).
P is the principal.
I is the yearly interest rate.
C is the number of conversions per year.
N is the number of years.

Input: Fields P, I, C, and N will be read from data statement. A nine in each field indicates you are finished processing.

Output: A five column table with appropriate names will be used to show the results of your processing.

FLOWCHART


START
INITIALIZE
PRINT COLUMN HEAD
R
READ P, I, C, N
ARE P,I,C, and N = 9
YES
STOP
NO
IS C = 0
PRINT DIVISION BY ZERO
R
NO
A - P 1+I/C NC
PRINT A, P, I, C, N
R

PROBLEM 2

SORTING A LIST OF NUMBERS

One of the most important functions of a computer in a data processing setting is to produce information in a form that makes it easily readable; that is, to place the information in a certain order. For now, we will concern ourselves with placing a list of numbers in ascending order (smallest to largest).

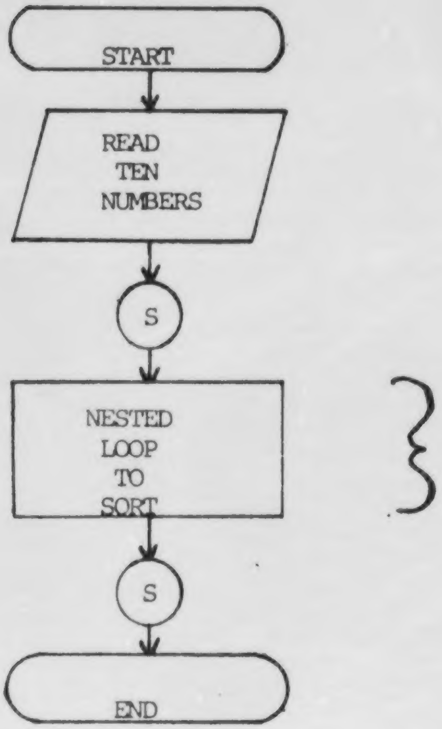
At the same time we can introduce "Subroutines" in BASIC. If a set of statements in a program is to be executed more than a few times at more than one place in a program, a subroutine can be used. This set of statements is called a subroutine.

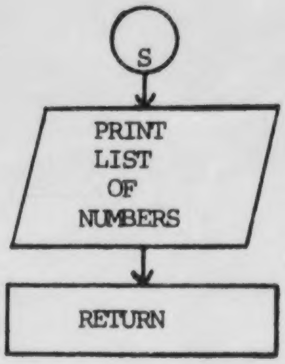
```
          .
          .
          .
 80 GO SUB 200
          .
          .
          .
200
210
220
230 RETURN
          .
          .
          .
999 END
```

Subroutine is statements 200-230. After statement 230 transfer is to statement after 80.

Be careful with statement before 200.

Now for the problem. All you have to do is read in ten numbers in random order (or let the computer generate them if you prefer) and print the numbers in ascending order.


START
READ TEN NUMBERS
S
NESTED LOOP TO SORT
S
END


S
PRINT LIST OF NUMBERS
RETURN

Each time a switch is made between any pair of numbers, use the subroutine to print the updated list.

Output should be formatted similar to the following.

```
     ORIGINAL LIST
         ...................................
     CHANGES AS SORTING IS DONE
         ...................................
         ...................................
         ...................................
     FINAL LIST - SMALLEST TO LARGEST
         ...................................
```

EVALUATION 02/07

Exchange Sort

In this problem you are going to write a program to do an exchange sort. For this problem we will limit the number of items to 15 two-digit numbers. You are to sort these 15 numbers in descending order.

Example: 5 Number Exchange Sort

Numbers: 10, 1, 98, 97, 4

Method:

1. Compare the first and second numbers.
2. If the second number is larger, exchange the two numbers; otherwise, leave them alone.
3. Compare the second and third numbers.
4. If the third number is larger than the second, exchange them; otherwise, leave them alone.

This method is continued until all of the items in the list have been compared. Then the entire sequence is begun again. One problem with this method is that you must make as many passes on the list as there are items in the list. A solution to this problem is the use of a switch. The switch is initially set at zero. Then, whenever an item is exchanged, a value of one is added to the switch. When the end of the list is reached, the switch is listed for a value of zero. If the switch is not zero, the switch is reset to zero and the entire list is sorted again. Let's look at how this works using the numbers in the example above.

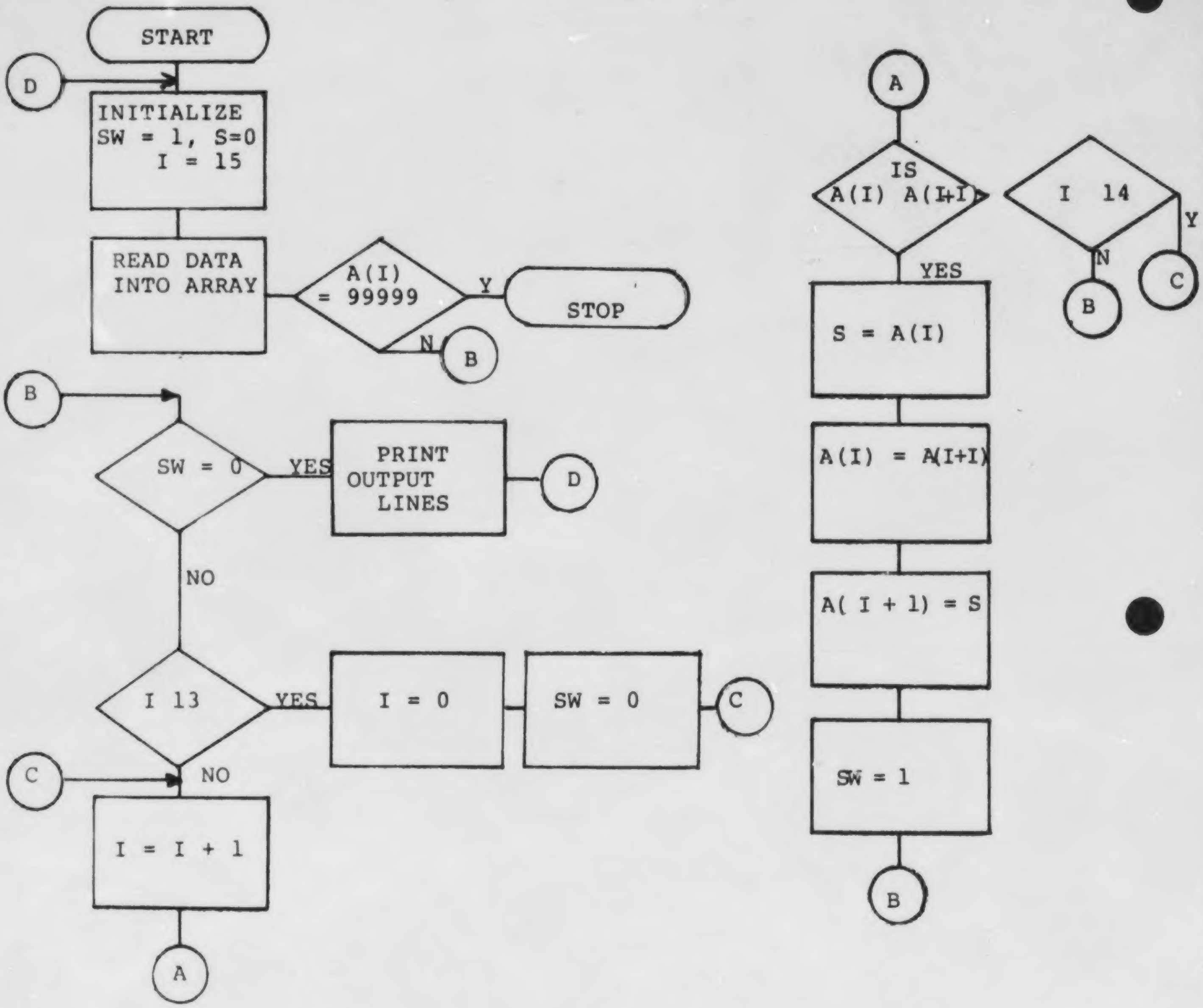
	Numbers	Switch
Before the first pass:	10, 1, 98, 97, 4	0
After the first pass:	10, 98, 97, 4, 1	1
After the second pass:	98, 97, 10, 4, 1	0

Since the switch is still zero, we know we are finished.

Input: Use the READ/DATA statements to obtain your input. There will be 15 sets of two-digit numbers.

Output: Print an appropriate heading for your list; skip a line, and print the numbers in descending order.

FLOWCHART


START
D
INITIALIZE
SW = 1, S=0
I = 15
READ DATA
INTO ARRAY
A(I)
= 99999
Y
STOP
N
B
B
SW = 0
YES
PRINT
OUTPUT
LINES
D
NO
I 13
YES
I = 0
SW = 0
C
C
NO
I = I + 1
A
A
IS
A(I) A(I+I)
YES
S = A(I)
A(I) = A(I+I)
A(I + 1) = S
SW = 1
B
I 14
Y
N
B
C

TEST DATA: Use at least three sets of data:

- one in random order
- one in ascending order
- one in descending order

Follow the last set of data with a 99999 card.

STUDENT COMPETENCY SHEET

MODULE 02: BASIC Programming

COMPETENCY 08: Write and run a BASIC program using IF-THEN, GOTO, and FOR-TO/NEXT statements.

PERFORMANCE GUIDE:

1. Write data descriptions, including uses of multiple loops and/or multiple branchings.
2. Review or revise flowchart in accordance with output requirements.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job program codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Review lessons in *Programming in BASIC, The Time-Sharing Language* on IF-THEN, GOTO, and FOR-TO/NEXT statements.

2. Review comparable sections in your manufacturer's BASIC reference manual.

3. Prepare and compile programs for the two problems in Handout 02/08/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's BASIC reference manual, data entry equipment, and coding sheets, prepare the required BASIC program and achieve an error-free compile within five computer runs.

HANDOUT 02/08/03

Problem: Electric Bill

In this problem you are going to calculate an electric bill for private resident accounts and commercial enterprise accounts. While you learn some more about programming, you will also learn how your parents' electric (and gas) bills are calculated. (The codes and rate steps used in this problem are not the ones used locally. If you want to obtain the correct codes and rate steps, the local gas and electric offices will supply them.)

To solve this problem you must use a subroutine to calculate the bills for both residential and commercial accounts. Your main-line program will read the data cards, call the proper subroutine, and print the calculated bill for each customer. When you finish processing the individual customers, skip 10 lines and print out a final billing report. The final billing report will show the number of customers by type (residential or commercial); the average kilowatt hours used, by type; and the average bill rendered for each type.

The rates you will use are as follows:

Rate Code	KWH Range	Rate Charged
1 (private resident)	50 KWH or less	$3.00 (minimum bill)
	51-150 KWH	$3.00 plus $ .05 per KWH over 50
	151-250 KWH	$8.00 plus $ .045 per KWH over 150
	251-350 KWH	$12.00 plus $.04 per KWH over 250
	Over 350 KWH	$16.50 plus $.035 per KWH over 350
2 (commercial enterprise)	1000 KWL or less	$35.00 (minimum bill)
	1001-5000 KWH	$35.00 plus $ .02 per KWH over 1000
	5001-20000 KWH	$135.00 plus $ .015 per KWH over 5000
	Over 20000 KWH	$360.00 plus $ .01 per KWH over 20000
3	Invalid Code - When an invalid (not valid) code is encountered, you are to flag the account number in which the error was detected by printing input data. Do not calculate a bill.	

Input: Rate schedule code, customer account number, previous meter reading, and current meter reading. A customer account number of 99999 signals the end of processing.

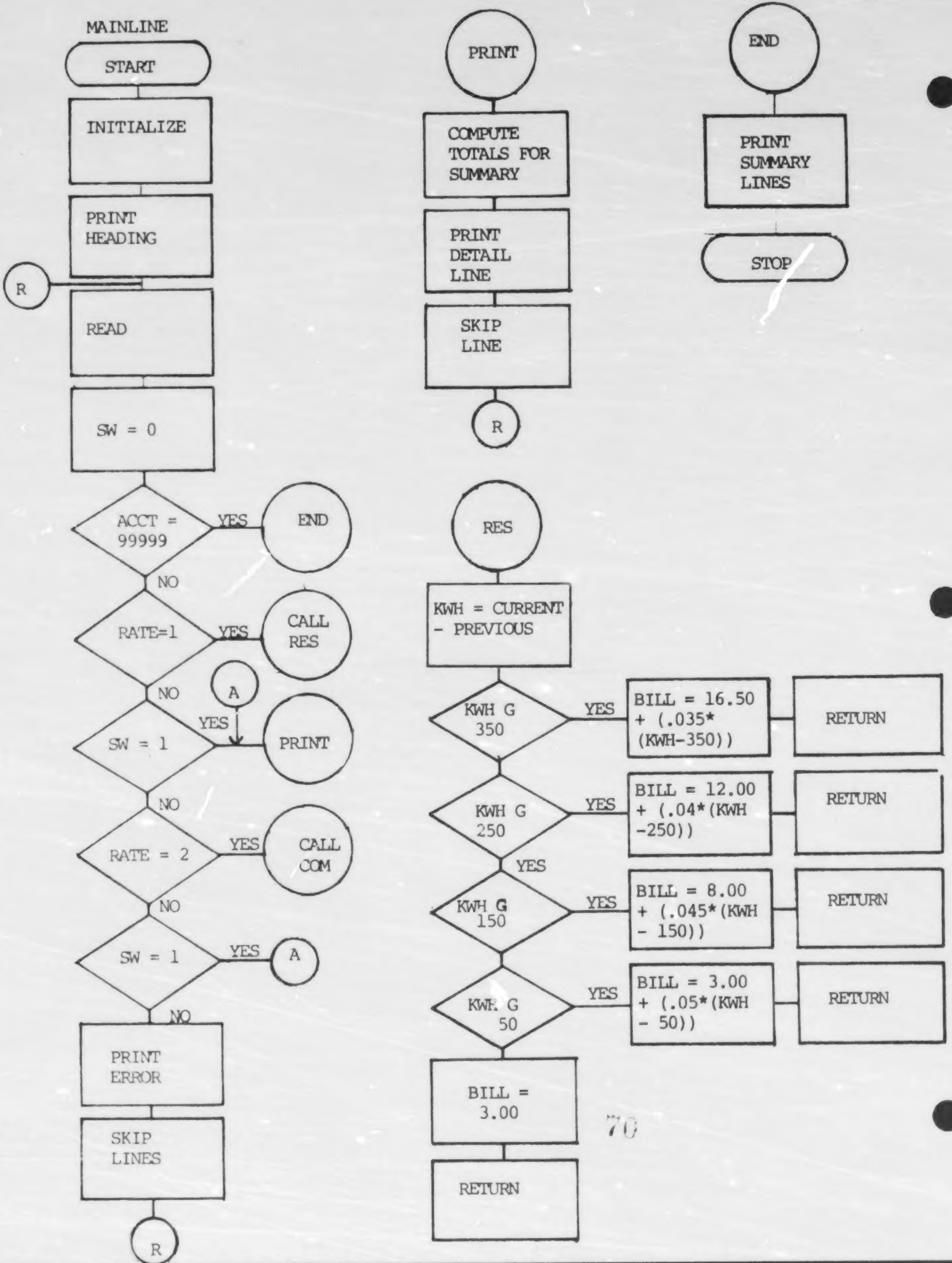
Output:

A. Customer bill for each customer (double-spaced) containing customer account number, rate code, previous meter reading, current meter reading, total units (kilowatts used), and amount of bill.

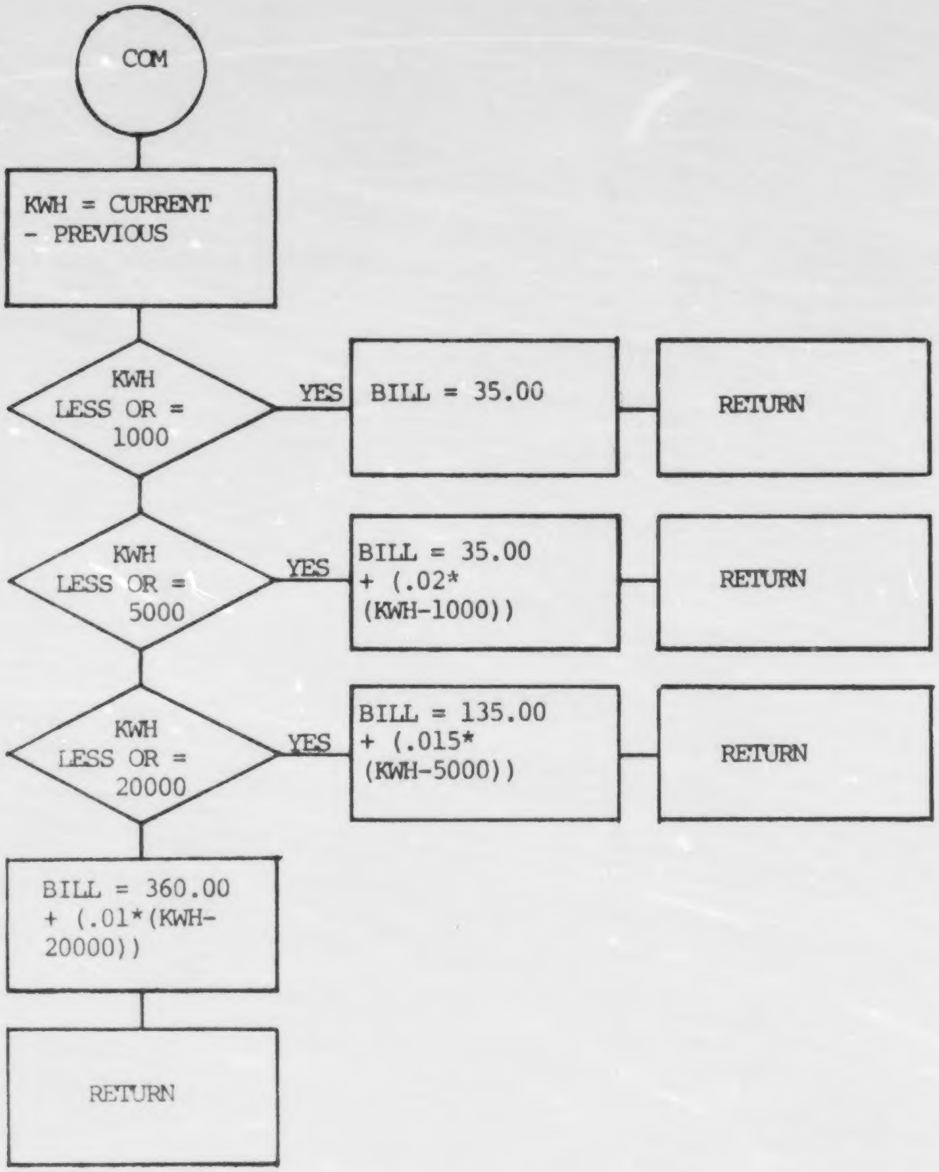
B. Error message on invalid rate codes showing account number, rate code, previous meter reading, and current meter reading.

C. Summary report.

FLOWCHART: This flowchart (see next page) contains several large "DO-IT" blocks. You are expected to fill them in.


MAINLINE
START
INITIALIZE
PRINT HEADING
R
READ
SW = 0
ACCT = 99999
YES
END
NO
RATE=1
YES
CALL RES
NO
A
SW = 1
YES
PRINT
NO
RATE = 2
YES
CALL COM
NO
SW = 1
YES
A
NO
PRINT ERROR
SKIP LINES
R
PRINT
COMPUTE TOTALS FOR SUMMARY
PRINT DETAIL LINE
SKIP LINE
R
END
PRINT SUMMARY LINES
STOP
RES
KWH = CURRENT - PREVIOUS
KWH G 350
YES
BILL = 16.50 + (.035*(KWH-350))
RETURN
KWH G 250
YES
BILL = 12.00 + (.04*(KWH -250))
RETURN
YES
KWH G 150
YES
BILL = 8.00 + (.045*(KWH - 150))
RETURN
KWH G 50
YES
BILL = 3.00 + (.05*(KWH - 50))
RETURN
BILL = 3.00
RETURN

FLOWCHART - page 2


COM
KWH = CURRENT - PREVIOUS
KWH LESS OR = 1000
YES
BILL = 35.00
RETURN
KWH LESS OR = 5000
YES
BILL = 35.00 + (.02* (KWH-1000))
RETURN
KWH LESS OR = 20000
YES
BILL = 135.00 + (.015* (KWH-5000))
RETURN
BILL = 360.00 + (.01*(KWH-20000))
RETURN

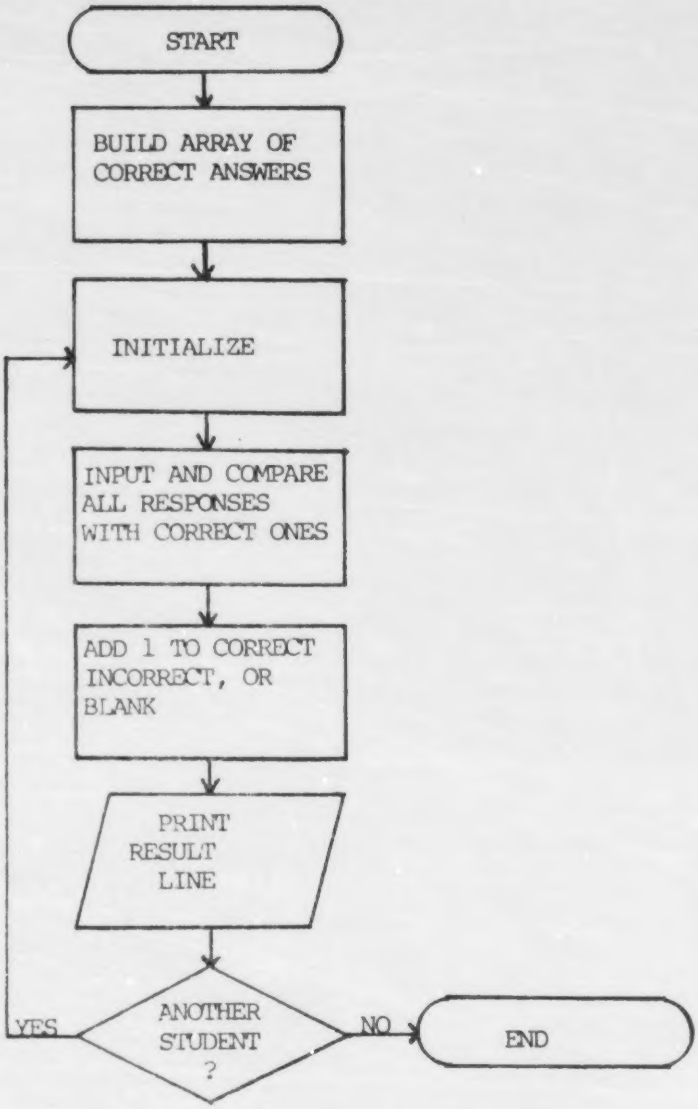
Problem 2 - Grading a Multiple Choice Test

This problem should be a challenge but you'll probably be very proud of yourself when it runs correctly. Let's assume there are 25 questions on a multiple-choice test. The possible answers are A, B, C, D. You'll have to use string variables to store the answers. Let's also say a blank will mean a question was not answered.

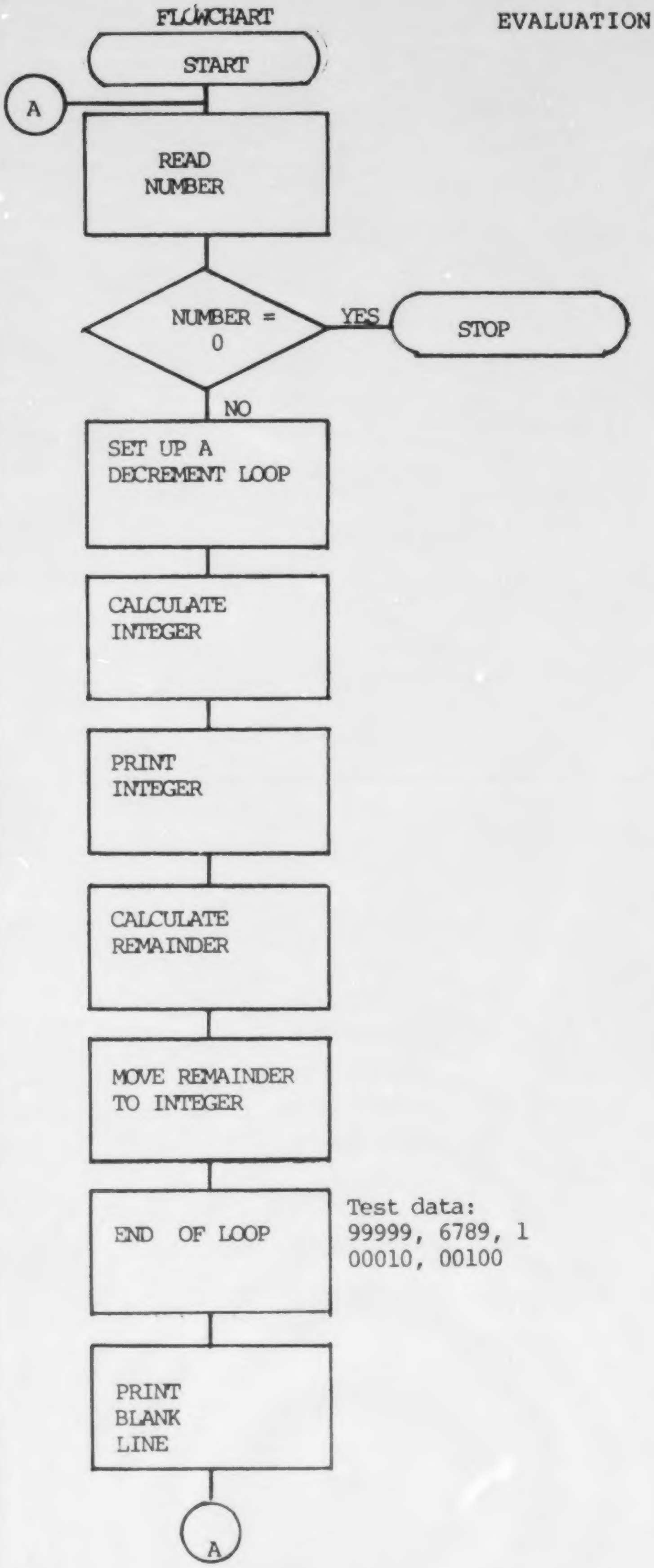
You are to write a program so that a typist can input a student name via the keyboard and then 25 responses for each student. Prompts should appear on the screen telling the typist to enter a name or an answer to a question.

When all input is complete for a student your program should count and output the number of correct answers, the number of incorrect answers, and the number of blank answers along with the student name. Remember, the computer will have to know the correct responses first. Data will be supplied for a number of students.

FLOWCHART:


START
BUILD ARRAY OF CORRECT ANSWERS
INITIALIZE
INPUT AND COMPARE ALL RESPONSES WITH CORRECT ONES
ADD 1 TO CORRECT INCORRECT, OR BLANK
PRINT RESULT LINE
ANOTHER STUDENT ?
YES
NO
END

EVALUATION 02/08


FLOWCHART
START
A
READ NUMBER
NUMBER = 0
YES
STOP
NO
SET UP A DECREMENT LOOP
CALCULATE INTEGER
PRINT INTEGER
CALCULATE REMAINDER
MOVE REMAINDER TO INTEGER
END OF LOOP
PRINT BLANK LINE
A

Test data:
99999, 6789, 1
00010, 00100

Problem: Digits of Integers

In this program you are to input an integer and print out the individual digits of the integer. There are three primary reasons for you to do this program: (1) to learn the uses of BASIC functions, (2) to learn more control of numbers for more complex problems, and (3) to practice the BASIC you have learned. For this problem an input value of zero signals when to stop.

Input: Any five-digit integer.

Output: On one line print the integer you read in. On the next line, print the individual digits of the integer.

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 01: Use required COBOL entries and correct COBOL syntax rules to achieve a compilation.

PERFORMANCE GUIDE:

1. From flow charts and problem statements, list data names for input, output, and work areas.
2. Code program on coding sheets.
3. Check coding for omissions and inconsistencies.
4. Enter program codes in preparation for compilation.
5. Proofread codes entered.
6. Enter job control codes in preparation for compilation.
7. Submit program for run.
8. Debug program using diagnostics.

LEARNING ACTIVITIES

1. Read Chapters 1-4 in COBOL Simplified. These are very short chapters. Answer the questions at the end of each chapter and have your answers checked.

2. Get two small, but complete COBOL programs from those available. Read what is coded in the IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, DATA DIVISION, and PROCEDURE DIVISION.

3. Read chapters 6-11 in COBOL Simplified. These are very short chapters. Answer the questions at the end of each chapter and have your answers checked.

4. Complete Handout 03/01/04. Have your answers checked.

5. Complete Handout 03/01/05. Have your answers checked.

6. Prepare and compile programs for the two problem statements in Handout 03/01/06. Use this procedure:

   a. Study the problem statements
   b. List data names to be used for input, output, and work areas.
   c. Code the programs on coding sheets.
   d. Enter the proofread program codes in preparation for compilation.
   e. Enter and proofread job control codes.
   f. Have program run.
   g. Debug program using diagnostics, until compilation is achieved.

EVALUATION

Given problem statement, flowchart, job control language, equipment reference manual, data entry equipment, and coding sheets, prepare the required COBOL program and obtain an error-free compile within three computer runs.

HANDOUT 03/01/04

Section I - Circle the correct answer

1. Alphabetic characters in COBOL are made up of:
    a. any character in the COBOL character set.
    b. A through Z #, ?, and blank.
    c. A through Z, blank.
    d. 0-9, A through Z.

2. The character set allowed for forming COBOL words consists of:
    a. any character in the COBOL character set.
    b. A through Z.
    c. first character numeric, followed by a space, followed by 0-9, - (hyphen).

3. The size limit on a COBOL word is:
    a. 30 characters.
    b. 8 characters.
    c. 16 digits.
    d. none of the above

4. A period used to end a statement must be followed by:
    a. a quotation mark.
    b. a space.
    c. 3 or more spaces.
    d. another character.
    e. none of the above

5. A non-numeric literal is bounded by:
    a. quotes.
    b. spaces.
    c. periods.
    d. none of the above.

6. A division name must begin:
    a. in column 1.
    b. in margin A.
    c. in margin B.
    d. anywhere between columns 12 and 72.

7. The continuation indicator is which character in which column?

    a. -, in column 73.
    b. *, in column 1.
    c. -, in column 7.
    d. any character, in column 7.

8. Which of the following is a legal COBOL WORD?
   a. NET-PAY OUT
   b. 91.568
   c. IN-OUT-95SUM
   d. CANE-BACK + CHAIR

9. Which of the following is not a legal figurative constant?
   a. QUOTE
   b. ZEROES
   c. ALL ZEROES
   d. SUMS

10. COBOL has ________________ divisions.
    a. 4
    b. 6
    c. 3 (Divisions) and 10 sections.
    d. no required

Section II - Indicate the needed changes in the coding. Rewrite:

```
     A    B
11.  |SOUR|CE - COMPUTER.  B5500, OBJECT-COMPUTER.  B5500
```

```
     A    B
12.  |PROG|RAM ID.  PAYROLL
```

```
     A    B
13.  |ENVI|RONMENT DIVISION CONFIGURATION SECTION.
```

```
     A    B
14.  |FD  |INPUT-FILE
           BLOCK CONTAINS 5
           VALUE OF ID IS 'BLIPBAP'.
```

```
     B
15.  |02 STUFF PICTURE 9V99 VALUE 'ABC'.
     B
16.  |02 JUNK PICTURE XXX VALUE 5.69.
     B
17.  |02 OTHER PICTURE XXX VALUE 123.
```

```
     A
18.  |DATA-DIVISION.
     FD FILE-1 etc.
```

```
     B
19.  |READ FILE-1.  MOVE REC-IN TO REC-WORK.
```

```
     B
20.  |ADD A, B, C, TO D GIVING E.
```

HANDOUT 03/01/04

ANSWERS

SECTION I

1. d
2. a
3. a
4. b
5. a
6. b
7. c
8. c
9. d
10. a

SECTION II

11. SOURCE COMPUTER. B5500. OBJECT COMPUTER. B5500
12. A period is required after PAYROLL
13. ENVIRONMENT DIVISION. CONFIGURATION SECTION.
14. BLOCK CONTAINS 5 RECORDS
15. XXX instead of 9V99
16. 9V99 instead of XXX
17. 999 instead of XXX
18. DATA DIVISION
19.
20. ADD A, B, C, D, GIVING E.

HANDOUT 03/01/05

Place an X beside each incorrect COBOL name and indicate the reason (s), listed below, by number.

REASONS

1. Length
2. Illegal beginning or ending
3. Imbedded spaces
4. No alpha character
5. Illegal character or word used

FILE-A
LOVE-IN
2ND-PART
AREA-2
ONCEUPONATIME
END-RUN
AZ
X-5
LAST-READ-TIME
ERROR-3
LINE 72-OUT
WORK A.
THIRD/TIME
TIME*OUT
4-WAY
DATA
2ND PART
A21
DIVISION
IDENTIFICATION
7-UP-
-SPECIAL-AREA
-2ND

-TRANSACT
-8TH-PART
"SIX"
CLEAR IT
3000
009
RECORD-2
WHOLE-THING
4-LINE
SECTION-20
TWENTYTHREESKIDOO
X
C-2-4
GO-BACK-TWICE
ONE/HALF
SAVE & STORE
AREA C
PRINT*OUT
6-21
REC ONE
WRITE-LINE 3
OVER-AND-OVER
SECTION

RECORD
PAY RATE
LAST-IN-
OUTPART-
A-3-
4-IN EXEC
HOME BASE
"3A"
ERR STOP
4215
68
PART*B
ROUTINE-FOR-CALCULATING-BONU[illegible]S
SECTION-NO.3
INVENTORY-UPDATING-SECTION
ERROR-ROUTINE
PARAGRAPH-142A-
R
AMT FLD
AREA-A/AREA-B
SECTION
3-B
CANCELLATION-OF-ACCOUNT-SECTION

HANDOUT 03/01/05

ANSWERS

Place an X beside each incorrect COBOL name and indicate the reason (s), listed below, by number.

REASONS

1. Length
2. Illegal beginning or ending
3. Imbedded spaces
4. No alpha character
5. Illegal character or word used

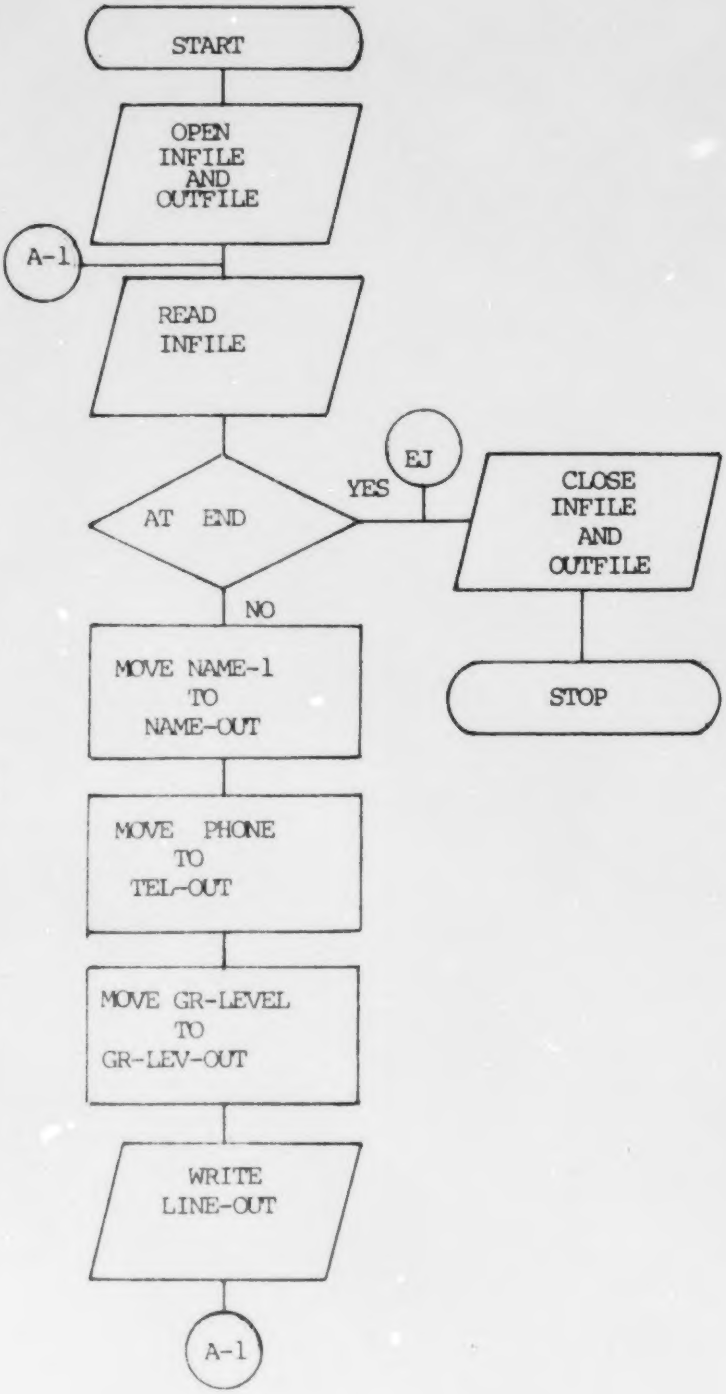
	FILE-A
	LOVE-IN
	2ND-PART
	AREA-2
	ONCEUPONATIME
	END-RUN
	AZ
	X-5
	LAST-READ-TIME
	ERROR-3
	LINE 72-OUT
X3,5	WORK A.
X5	THIRD/TIME
X3	TIME*OUT
	4-WAY
X5	DATA
X3	2ND PART
	A21
X5	DIVISION
X5	IDENTIFICATION
X2	7-UP-
X2	-SPECIAL-AREA
X2	-2ND
X2	-TRANSACT
X2	-8TH-PART
X5	"SIX"
X3	CLEAR IT
X4	3000
X4	009
	RECORD-2
	WHOLE-THING
	4-LINE
	SECTION-20
	TWENTYTHREESKIDOO
	X
	C-2-4
	GO-BACK-TWICE
X5	ONE/HALF
X3,5	SAVE & STORE
X3,5	AREA C
X5	PRINT*OUT
X4	6021
X3	REC ONE
X3	WRITE-LINE 3
	OVER-AND-OVER
X5	SECTION
X5	RECORD
X3	PAY RATE
X2	LAST-IN-
X2	OUTPART-
X2	A-3-
X3	4-IN EXEC
X3	HOME BASE
X5	"3A"
X3	ERR STOP
X4	4215
X4	68
X5	PART*B
X1	ROUTINE-FOR-CALCULATING-BONUSES
X5	SECTION-NO.3
	INVENTORY-UPDATING-SECTION
	ERROR-ROUTINE
X2	PARAGRAPH-142A-
	R
X3	AMT FLD
X5	AREA-A/AREA-B
X5	SECTION
	3-B
X1	CANCELLATION-OF-ACCOUNT-SECTION

HANDOUT 03/01/06

PROBLEM STATEMENT 1

STUDENT LIST

FLOWCHART


START
OPEN INFILE AND OUTFILE
A-1
READ INFILE
AT END
YES
EJ
CLOSE INFILE AND OUTFILE
STOP
NO
MOVE NAME-1 TO NAME-OUT
MOVE PHONE TO TEL-OUT
MOVE GR-LEVEL TO GR-LEV-OUT
WRITE LINE-OUT
A-1

STATEMENT

The Student Council of Gung-Ho High School wants to publish a list of students so that council members can contact other students by phone.

Cards containing names, phone numbers, and grade levels are available; these are all to be printed with blank spaces between each field on the printout.

Input: CARDS

NAME	1-24
PHONE	25-32
GR-LEVEL	33-34
SEX	35
BLANK	36-80

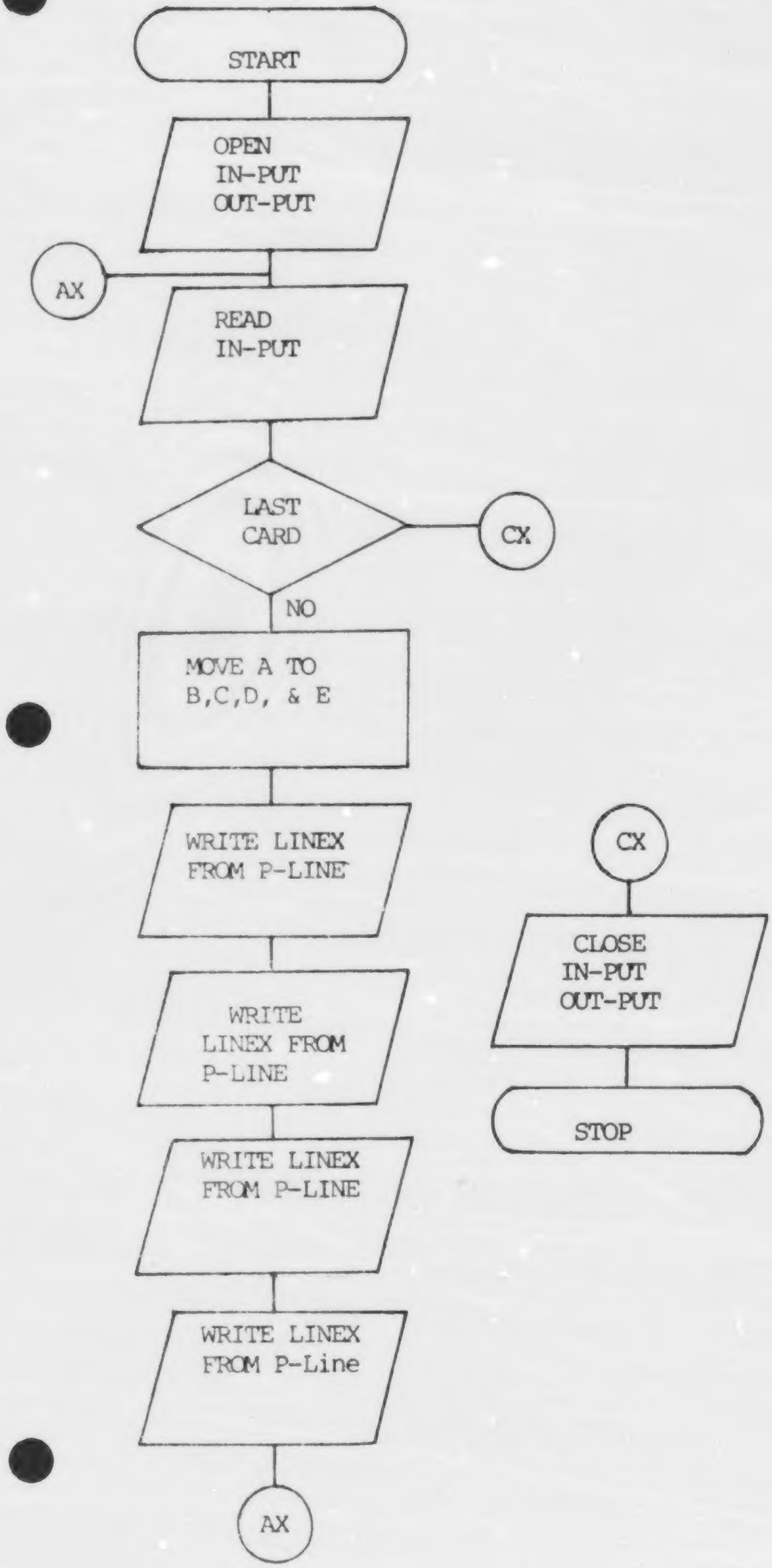
Output: PRINTER

BLANK	1-25
NM-OUT	26-49
BLANK	50-56
TEL-OUT	57-64
BLANK	65-73
GR-LEV-OUT	74-75
BLANK	76-132

PROBLEM STATEMENT 2

SIMPLE DESIGN

FLOWCHART


START
OPEN
IN-PUT
OUT-PUT
AX
READ
IN-PUT
LAST
CARD
CX
NO
MOVE A TO
B,C,D, & E
WRITE LINEX
FROM P-LINE
WRITE
LINEX FROM
P-LINE
WRITE LINEX
FROM P-LINE
WRITE LINEX
FROM P-Line
AX
CX
CLOSE
IN-PUT
OUT-PUT
STOP

STATEMENT

There are many ways to print complex layouts and designs on the computer. The simplest way is by using a data card for each line.

In this problem each data card is used to print several lines.

Input:

```
    DATA-CARD
       A      PIC X(16)
       FILLER PIC X(64)
```

Output:

```
    LINEX PIC X(132)
```

Working-Storage:

```
    P-LINE

       FILLER PIC X(29)
       B PIC X(16)
       C PIC X(16)
       D PIC X(16)
       E PIC X(16)
       FILLER PIC X(29)
```

EXERCISE 2

ANSWERS

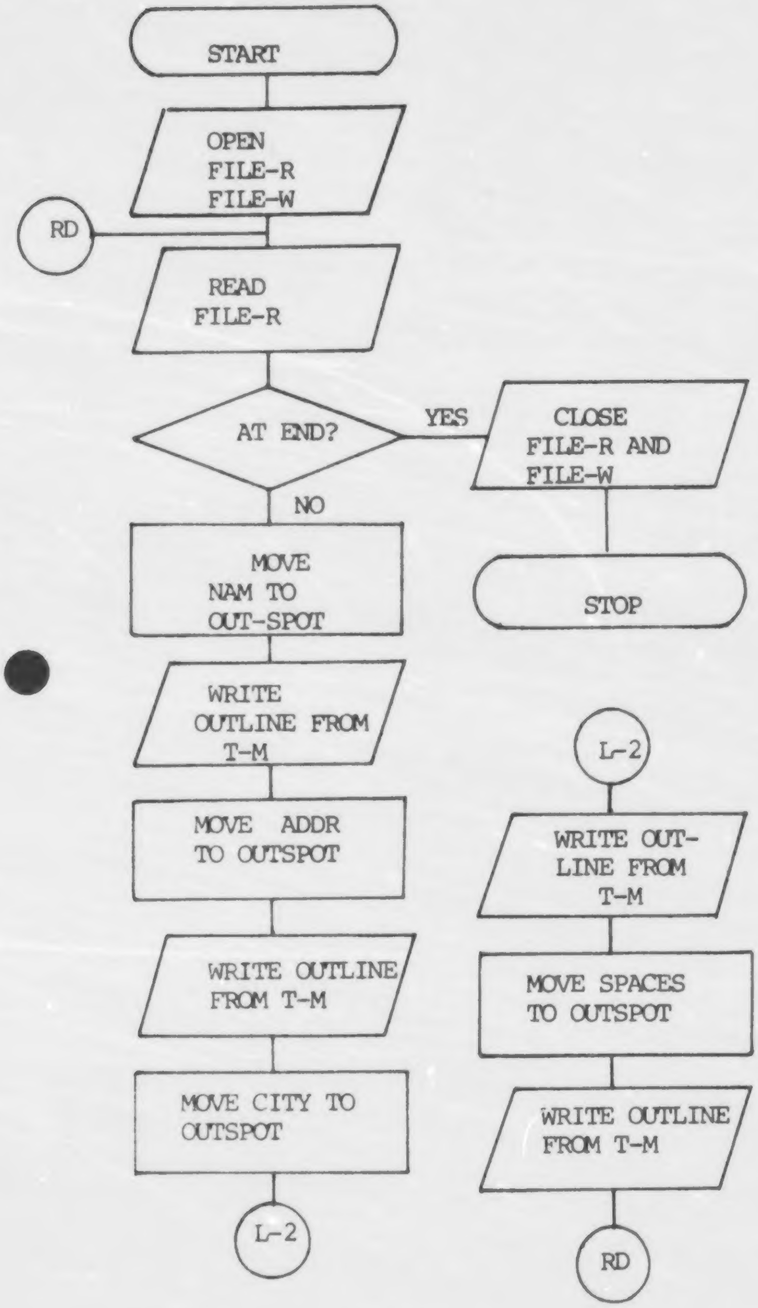
```
    B
1.  MOVE 1 TO AREA-C.
    GO TO R-X

    B
2.  IF CHUCK NUMERIC
    GO TO TR, ELSE GO TO CP.

    B
3.  IF A = B GO TO STEP 3
    ELSE ADD 1 TO A.

    B
4.  IF TOM NOT EQUAL HARRY MOVE
    ZERO TO PLACE, GO TO ALASKA ELSE
    GO TO TEXAS

    B
5.  IF M=3 ADD 1 TO CTR.
    GO TO BB.

    B
6.  IF WK-1  SAVE GO TO 2A ELSE IF
    WK-1  SAVE GO TO 2C ELSE GO TO
    2B

    B
7.  IF X NOT GREATER THAN Y GO TO 4B.
    IF X NOT GREATER THAN Z GO TO 12C.
    GO TO 17A.

    B
8.  IF T-CD NOT = 1 NEXT SENTENCE ELSE
       IF SW = D?
    IF T-CD NOT = 2 NEXT SENTENCE ELSE?
    IF T-CD NOT = 3 GO TO I4 ELSE?
```

EVALUATION 03/01

PROBLEM STATEMENT

NAMES AND ADDRESSES

FLOWCHART


START
OPEN FILE-R FILE-W
RD
READ FILE-R
AT END?
YES
NO
CLOSE FILE-R AND FILE-W
STOP
MOVE NAM TO OUT-SPOT
WRITE OUTLINE FROM T-M
MOVE ADDR TO OUTSPOT
WRITE OUTLINE FROM T-M
MOVE CITY TO OUTSPOT
L-2
L-2
WRITE OUT-LINE FROM T-M
MOVE SPACES TO OUTSPOT
WRITE OUTLINE FROM T-M
RD

STATEMENT

The Advertising has decided to use some direct mail ads, and wishes the Computer Department to print some address labels. All of the necessary information is punched into cards.

Input:

```
    Add-CARD
       NAME PIC X(25)
       ADDR PIC X(25)
       CITY PIC X(25)
       FILLER PIC X(5)
```

Output:

```
    OUTLINE PIC X(132)
```

Working Storage:

```
    T-M
       FILLER PIC X(10)
       OUTSPOT PIC X(25)
       FILLER PIC X(97)
```

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 02: Prepare input, output, and work file descriptions for COBOL program requirements.

PERFORMANCE GUIDE:
1. From flowcharts and problem statements, list data names for input, output, and work areas.
2. Develop and write data descriptions.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compilation.
6. Proofread codes entered.
7. Enter job control codes in preparation for compilation.
8. Submit program for run.
9. Debug program using diagnostics.

LEARNING ACTIVITIES

1. Read Chapters 13-19 in COBOL Simplified. These are very short chapters. Answer the questions at the end of each chapter and have your answers checked.

2. Select a card, layout sheet, or a sheet showing a record on tape or disc from those available. Write a description for the card, layout sheet, or sheet and an FD statement.

3. Read Chapters 22-26 in COBOL Simplified. These are very short chapters. Answer the questions at the end of each chapter and have your answers checked.

4. Prepare and compile programs for the two problem statements in Handout 03/02/04. Use this procedure:
    a. Study the problem statements.
    b. List data names to be used for input, output, and work areas.
    c. Write data descriptions.
    d. Code the programs on coding sheets.
    e. Enter and proofread program codes in preparation for compilation.
    f. Enter and proofread job control codes.
    g. Have program run.
    h. Debug program using diagnostics until compilation is achieved.

EVALUATION

Given problem statement, flowchart, job control language, equipment reference manual, data entry equipment, and coding sheets, prepare the required COBOL program and obtain an error-free compile within three computer runs.

HANDOUT 03/02/04

PROBLEM STATEMENT 1

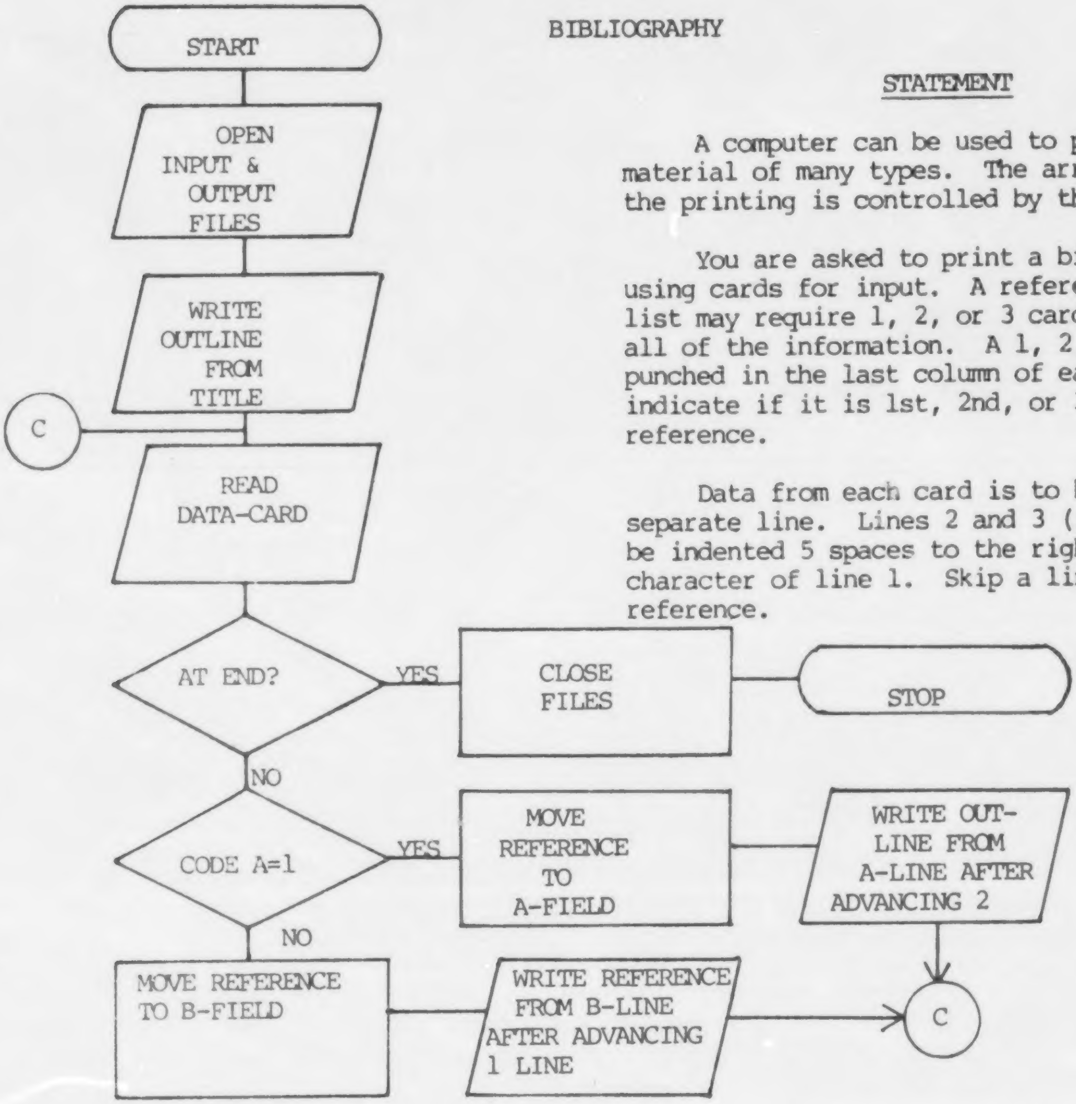
BIBLIOGRAPHY

STATEMENT

A computer can be used to print text material of many types. The arrangement of the printing is controlled by the program.

You are asked to print a bibliography, using cards for input. A reference on the list may require 1, 2, or 3 cards to contain all of the information. A 1, 2, or 3 is punched in the last column of each card to indicate if it is 1st, 2nd, or 3rd for a reference.

Data from each card is to be printed on a separate line. Lines 2 and 3 (if any) are to be indented 5 spaces to the right of the 1st character of line 1. Skip a line between each reference. (This can be done by using the "AFTER ADVANCING" clause on the WRITE statement each time you write LINE 1.)


START
OPEN INPUT & OUTPUT FILES
WRITE OUTLINE FROM TITLE
C
READ DATA-CARD
AT END?
YES
CLOSE FILES
STOP
NO
CODE A=1
YES
MOVE REFERENCE TO A-FIELD
WRITE OUT-LINE FROM A-LINE AFTER ADVANCING 2
NO
MOVE REFERENCE TO B-FIELD
WRITE REFERENCE FROM B-LINE AFTER ADVANCING 1 LINE
C

Input:

1-79 REFERENCE
80 CODE-A

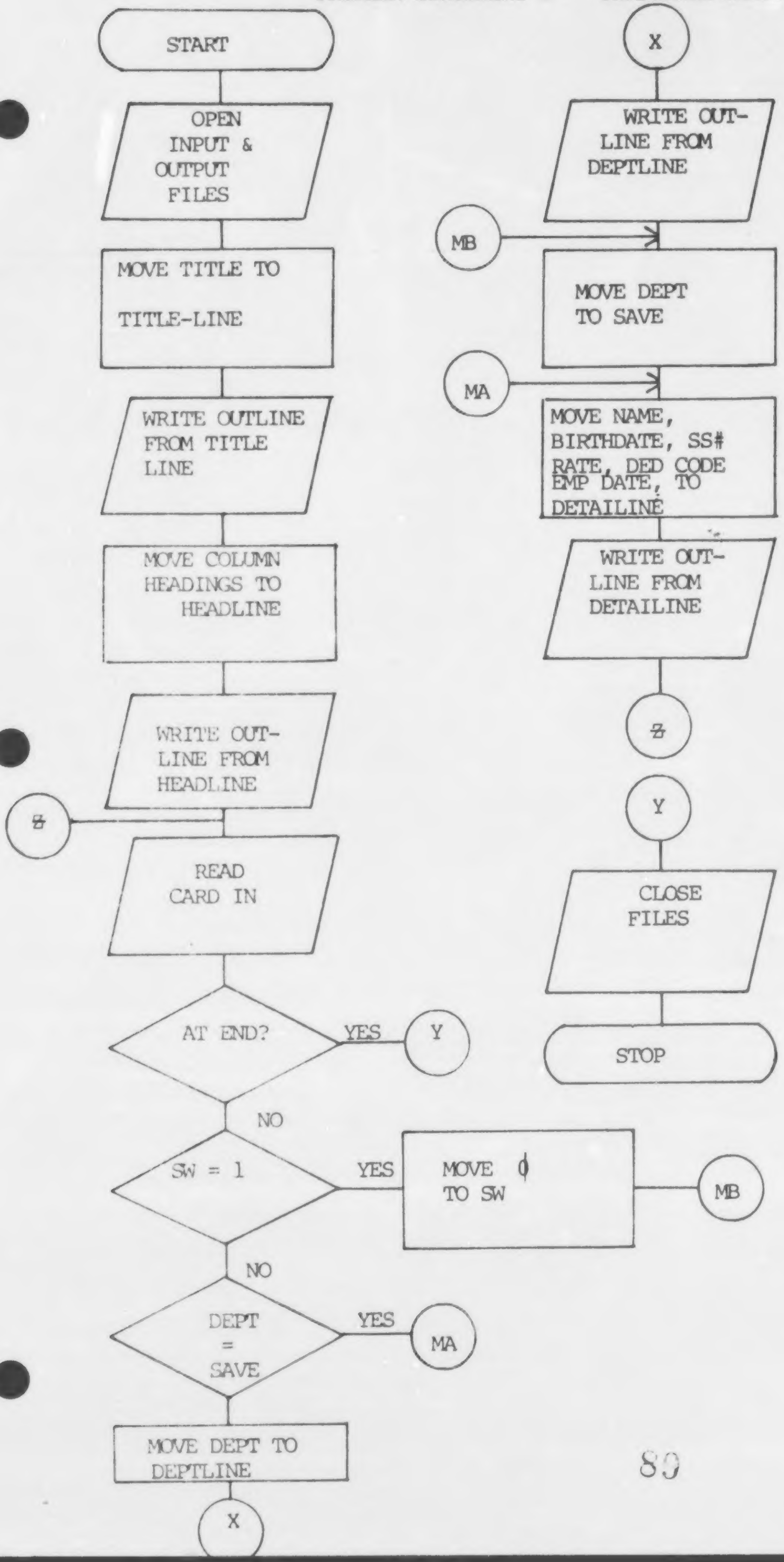
Output:

1. Print a heading - "BIBLIOGRAPHY"
2. Detail Line - center the 85-column printed area in the 132 printline.

Working-Storage:
In order to indent the data from the cards on the 2nd and 3rd lines, set up 2 separate work areas of 132 columns each.
Example:

```
01 A-LINE
   02 FILLER PIC X(23).
   02 A-FIELD PIC X(80).
   02 FILLER PIC X(29).

01 B-Line.
   02 FILLER PIC X(28).
   02 B-FIELD PIC X(80).
   02 FILLER PIC X(24).
```

PROBLEM STATEMENT 2 -- PERSONNEL RECORD


START
OPEN INPUT & OUTPUT FILES
MOVE TITLE TO TITLE-LINE
WRITE OUTLINE FROM TITLE LINE
MOVE COLUMN HEADINGS TO HEADLINE
WRITE OUT-LINE FROM HEADLINE
Z
READ CARD IN
AT END?
YES
Y
NO
SW = 1
YES
MOVE 0 TO SW
MB
NO
DEPT = SAVE
YES
MA
MOVE DEPT TO DEPTLINE
X
X
WRITE OUT-LINE FROM DEPTLINE
MB
MOVE DEPT TO SAVE
MA
MOVE NAME, BIRTHDATE, SS# RATE, DED CODE EMP DATE, TO DETAILINE
WRITE OUT-LINE FROM DETAILINE
Z
Y
CLOSE FILES
STOP

STATEMENT

The Personnel Department has asked you to provide a list of all employees showing information from the Master File. You are asked to provide a title and column headings.

Employees are grouped by department and the department number is to be printed once at the beginning of each department.

Make a printer layout before coding the program.

Input:

1-5	Emp. No.
6-14	Soc. Sec. #
15-28	Name
29-34	Birthdate
35-40	Emp-date
41-44	Pay Rate
45-46	Deduction Code
47-77	Blank
78-80	Dept

Output:

TITLE-LINE
Center "PERSONNEL RECORD"

HEADLINE
Use the following headings, to be printed over each print field: "NO", "NAME", "SOC. SEC. NO." "RATE", "BIRTHDATE", "EMPLOYMENT DATE", and "DEDUCTIONS".

(Note: For a well balanced report, plan the spacing of the detail lines first; then arrange the field headings evenly above each field. If necessary, headings can be abbreviated.)

PERSONNEL RECORD - continued

DETAILINE

Fields should be printed in the order indicated for the HEADLINE. Spaces between fields and the left and right margins should all be approximately the same.

Hint: Calculate the total columns needed for the printed fields. These can be determined from the input description (except that the Social Security number should be divided into 3 groups and printed with spaces separating the groups):

```
                02 SOC-SEC.
                   03 S-1 PIC XXX.
(Input)            03 S-2 PIC XX.
                   03 S-3 PIC XXXX.

                02 SOC-SEC-OUT
                   03 S-4 PIC XXX.
                   03 FILLER PIC X.
                   03 S-5 PIC XX.
                   03 FILLER PIC X.
                   03 S-6 PIC XXXX.
```

Subtract total count from 132. Divide the difference by 8 (the number of fields plus 1). This gives you the number of blank spaces available between fields. You may need to make adjustments if the quotient is a fraction.

DEPTLINE

Center "DEPARTMENT XXX" on the line. (XXX represents a field into which the actual number of the department is to be (inserted.)

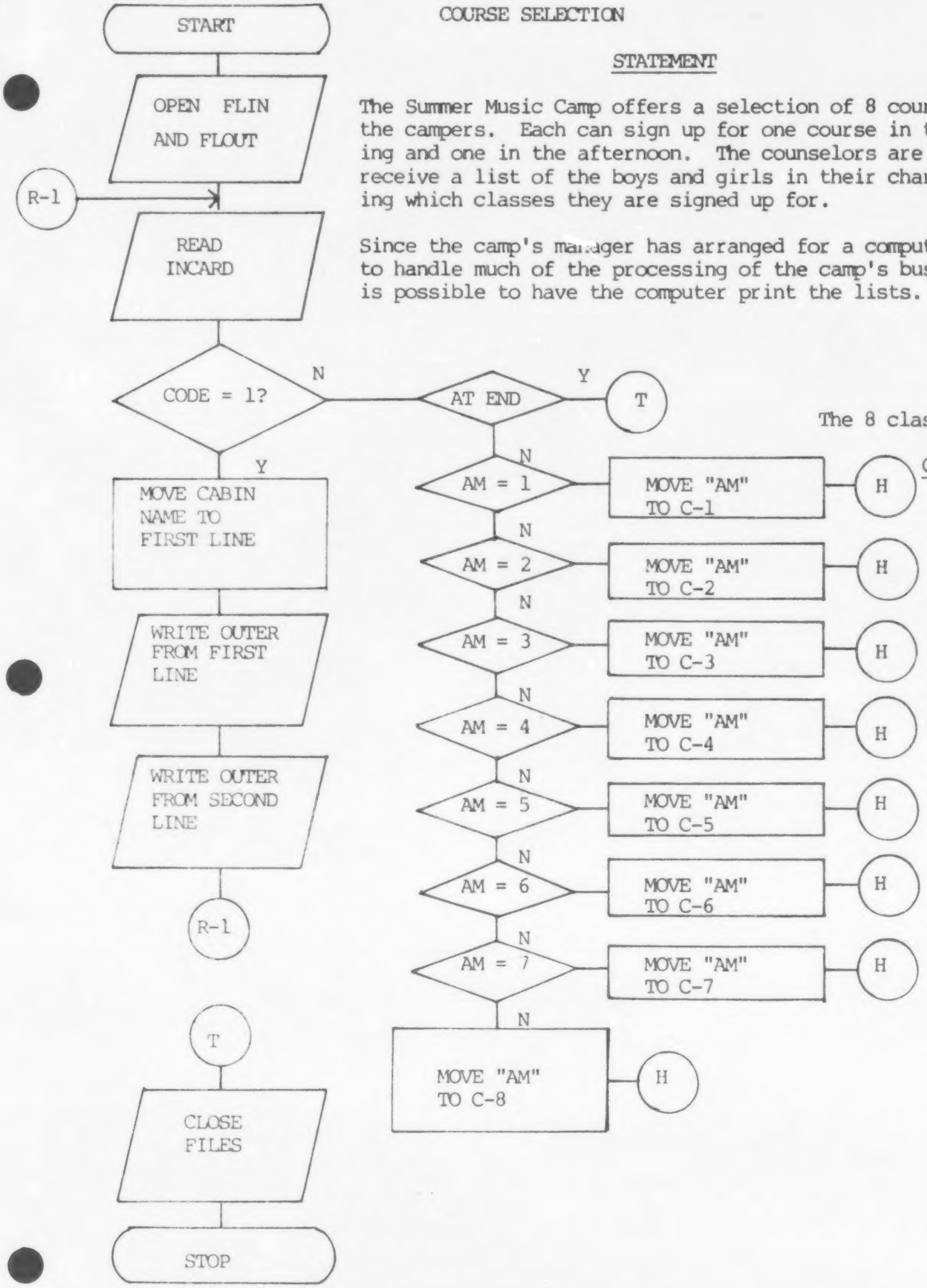
COURSE SELECTION

STATEMENT

The Summer Music Camp offers a selection of 8 courses for the campers. Each can sign up for one course in the morning and one in the afternoon. The counselors are each to receive a list of the boys and girls in their charge showing which classes they are signed up for.

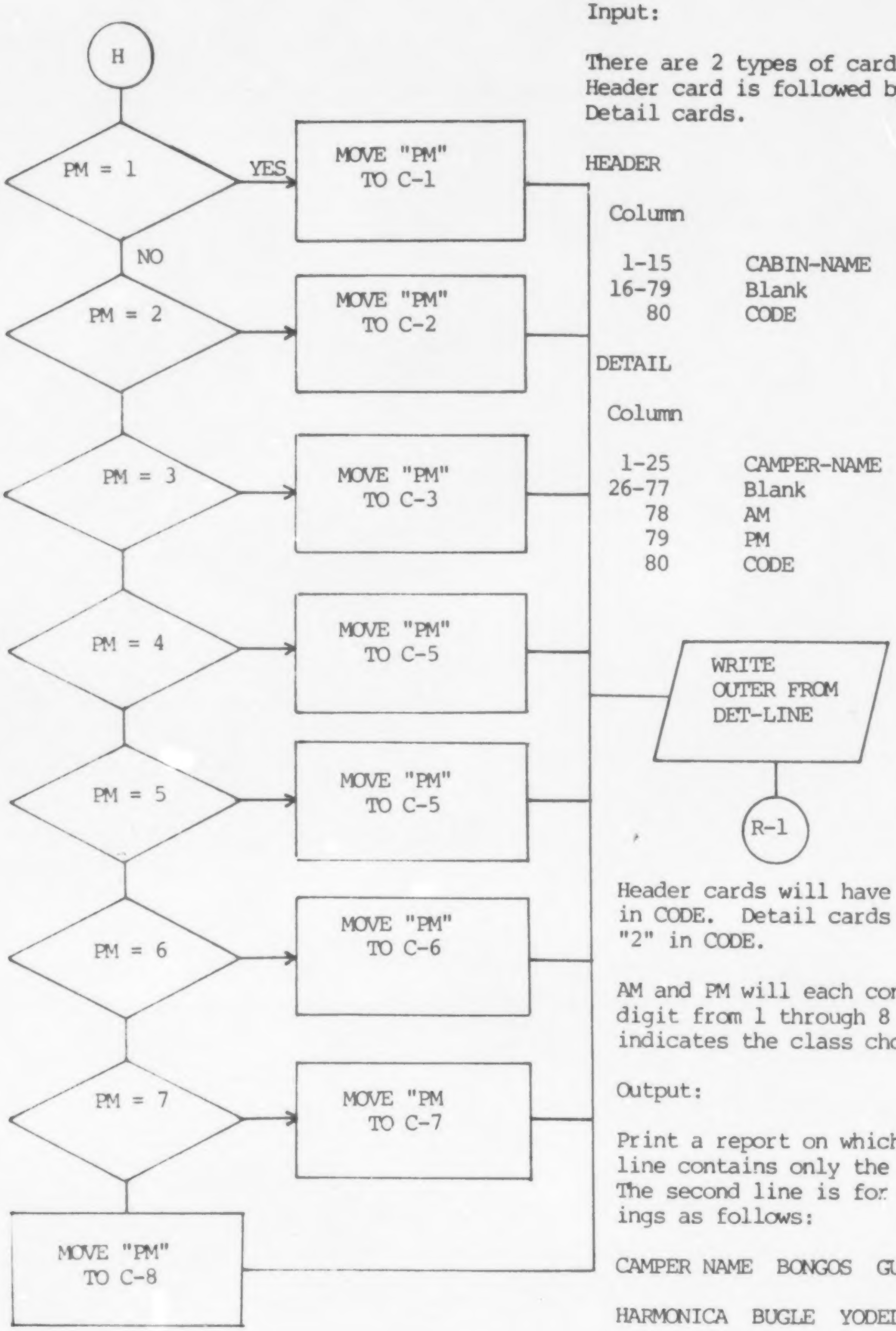
Since the camp's manager has arranged for a computer service to handle much of the processing of the camp's business, it is possible to have the computer print the lists.

FLOWCHART


START
OPEN FLIN AND FLOUT
R-1
READ INCARD
CODE = 1?
N
Y
AT END
Y
T
MOVE CABIN NAME TO FIRST LINE
WRITE OUTER FROM FIRST LINE
WRITE OUTER FROM SECOND LINE
R-1
N
AM = 1
MOVE "AM" TO C-1
H
N
AM = 2
MOVE "AM" TO C-2
H
N
AM = 3
MOVE "AM" TO C-3
H
N
AM = 4
MOVE "AM" TO C-4
H
N
AM = 5
MOVE "AM" TO C-5
H
N
AM = 6
MOVE "AM" TO C-6
H
N
AM = 7
MOVE "AM" TO C-7
H
N
MOVE "AM" TO C-8
H
T
CLOSE FILES
STOP

The 8 classes are:

Code	Name
1	Bongos
2	Guitar
3	Harmonica
4	Bugle
5	Yodeling
6	Whistling
7	Zither
8	Tambourine


H
PM = 1
YES
NO
MOVE "PM" TO C-1
PM = 2
MOVE "PM" TO C-2
PM = 3
MOVE "PM" TO C-3
PM = 4
MOVE "PM" TO C-5
PM = 5
MOVE "PM" TO C-5
PM = 6
MOVE "PM" TO C-6
PM = 7
MOVE "PM TO C-7
MOVE "PM" TO C-8
WRITE OUTER FROM DET-LINE
R-1

Input:

There are 2 types of cards. A Header card is followed by several Detail cards.

HEADER

Column	
1-15	CABIN-NAME
16-79	Blank
80	CODE

DETAIL

Column	
1-25	CAMPER-NAME
26-77	Blank
78	AM
79	PM
80	CODE

Header cards will have a "1" punched in CODE. Detail cards will have a "2" in CODE.

AM and PM will each contain a digit from 1 through 8 which indicates the class choice.

Output:

Print a report on which the first line contains only the cabin name. The second line is for column headings as follows:

CAMPER NAME BONGOS GUITAR

HARMONICA BUGLE YODELING ETC.

The detail lines will each contain the name of a camper and an "AM" and "PM" beneath one of the class headings.

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 03: Use diagnostic messages for efficient correction of COBOL syntax errors.

PERFORMANCE GUIDE:

1. Review diagnostic messages against data names and descriptions.
2. Check diagnostic messages against manufacturer's COBOL reference manual.
3. Correct program coding errors based on diagnostic messages.
4. Submit program for rerun.
5. Debug program further using subsequent diagnostic messages.

LEARNING ACTIVITIES

1. Each computer manufacturer has its own COBOL compiler which produces diagnostic messages as output. Get your manufacturer's COBOL reference manual and review the diagnostic messages listed in it.

2. Attend a group discussion on the diagnostic messages listed in your manufacturer's COBOL reference manual. Make a list of the most common COBOL syntax errors. Also answer this question: Does the number of diagnostic messages have any relationship to the number of coding errors? Why or why not?

3. From the COBOL programs available in your classroom, get a program that has been run and has diagnostic messages. For each message write a probable coding error. Have your work checked.

4. From the COBOL programs available in your classroom, get a program that has been run and has diagnostic messages. Debug the program using this procedure:
   a. Check data names and descriptions.
   b. Check diagnostic messages against manufacturer's COBOL reference manual.
   c. Correct program coding errors.
   d. Have program rerun.
   e. Debug program further.

EVALUATION

Given COBOL program with diagnostic messages, manufacuturer's COBOL reference manual, data entry equipment, and coding sheets, correct the syntax errors and obtain an error-free compile within two computer runs.

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 04: Perform data handling and manipulations by use of COBOL conditional statements and COBOL arithmetic statements.

PERFORMANCE GUIDE:

1. From flowcharts and problem statements, list data names for input, output, and work areas.
2. Develop and write data descriptions.
3. Finalize flowcharts and complete layouts and other documentation.
4. Code program on coding sheets.
5. Check coding for omissions and inconsistencies.
6. Enter program codes in preparation for compilation.
7. Proofread codes entered.
8. Enter job control codes in preparation for compilation.
9. Submit program for run.
10. Debug program using diagnostics.
11. Test program using test data.

LEARNING ACTIVITIES

1. Complete the following reading assignments. Each time answer the questions at the end of the chapter in COBOL Simplified and then review the same topic in your manufacturer's reference manual. Have your answers checked.
   a. Read Chapters 20 and 21 on COMPUTATIONAL statements.
   b. Read Chapter 27 on PROCEDURE DIVISION.
   c. Read Chapters 32-36 and 42 on ADD, SUBTRACT, MULTIPLY, DIVIDE, COMPUTE, GO TO, and IF.

2. Complete Handout 03/04/02. Have your answers checked.

3. Prepare and compile programs for the two problem statements in Handout 03/04/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, equipment reference manual, data entry equipment, and coding sheets, prepare the required COBOL program and obtain an error-free compile within five computer runs.

HANDOUT 03/04/02

EXERCISE 1

A. Use the ADD, SUBTRACT, MULTIPLY OR DIVIDE verbs to put the following statements into COBOL.

1. $A + B = C$

2. $A + B = A$

   An equation like this wouldn't be acceptable in your algebra class, but to the computer it means: add field A and field B, and store the results in field A.

3. What happened to the original data in field A? ______________

4. What happened to the original data in field B? ______________

5. $R = M + N + O + P = Q$

6. $M - N = A$

7. $Y - X - 42$

8. $(C)\ (D) = E$

9. $M = 5\ (1)$

10. $R = \frac{P}{Q}$

11. $S = \frac{S}{A}$

12. $C/4 = D$

B Use the COMPUTE statement to translate the following statements into COBOL.

13. $T = (A + B)\ \ (A + C)$

14. $A + B + C + D = E$

15. $X = \frac{3RS}{(S-4)}$

16. Dog = Head + Tail/4 (Legs)

17. $\mathrm{DELTA} = \frac{3\ (A + R)\ \ (A + S)}{S\ (\frac{R}{3}) + \frac{S-3}{4A}}$

18. $T = 4 = (3 \cdot 8 \cdot 9)\ /\ [5/7 - 16]$

19. $X^2 + Y^3 - 2Y^2 = Z$

20. $R - A + A^2\ (3A) - (2A^3)\ (\frac{5A}{2})$

HANDOUT 03/04/02

EXERCISE 1

(ANSWERS)

A. Use the ADD, SUBTRACT, MULTIPLY OR DIVIDE verbs to put the following statements into COBOL.

1. $A + B = C$ Add A, B Giving C,

2. $A + B = A$ Add B to A

   An equation like this wouldn't be acceptable in your algebra class, but to the computer it means: add field A and field B, and store the results in field A.

3. What happened to the original data in field A? increased by B

4. What happened to the original data in field B? Nothing

5. $R = M + N + O + P = Q$ two "=" signs..., an invalid condition.

6. $M - N = A$ Subtract N from M giving A.

7. $Y - X - 42$ Subtract x, 42 from Y giving?

8. $(C)\ (D) = E$ Multiply C, D Giving E

9. $M = 5\ (1)$ Multiply 5, 1 Giving M.

10. $R = \frac{P}{Q}$ Divide P by Q giving R,

11. $S = \frac{S}{A}$ Divide S by A

12. $C/4 = D$ Divide C by 4 giving D

B Use the COMPUTE statement to translate the following statements into COBOL.

13. $T = (A + B)\ (A + C)$ Compute T = (A+B) * (A+C)

14. $A + B + C + D = E$ Compute E = (A+B+C+D)

15. $X = \frac{3RS}{(S-4)}$ Compute X = (3*R*S) / (S-4)

16. $\text{Dog} = \text{Head} + \text{Tail}/4\ (\text{Legs})$ Compute Dog = Head + (Tail/ (4Legs))

17. $\text{DELTA} = \frac{3\ (A + R)\ \ (A + S)}{S\ \left(\frac{R}{3}\right) + \frac{S-3}{4A}}$ Compute Delta = (3*(A+R)*(A+S))/((S*R/3)+((S-3)/(4*A)))

18. $T = 4 = (3 \cdot 8 \cdot 9)\ /\ [5/7 - 16]$ Unresolvable - 2 "=" signs

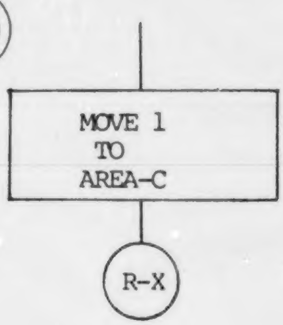
19. $X^2 + Y^3 - 2Y^2 = Z$ Compute Z = X**Z + Y**3 - 2*Y**2

20. $R - A + A^2\ (3A) - (2A^3)\ \left(\frac{5A}{2}\right)$ Compute ? = R-A+A**2*3*A -(2*A**3)*(5*A/2)

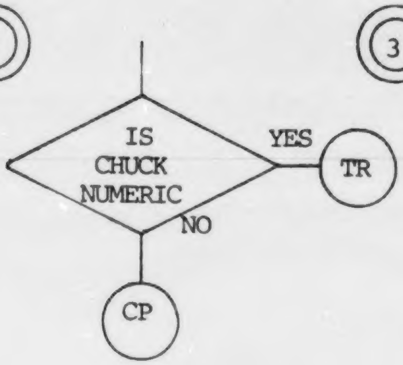
EXERCISE 2

Write COBOL statements for the following flowchart segments:

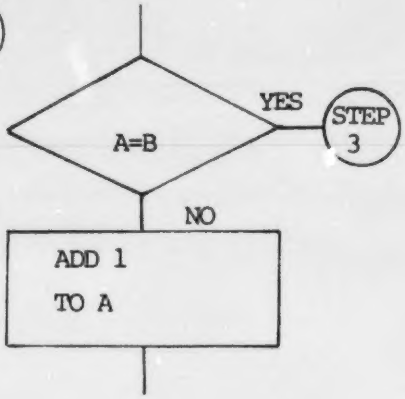
①


MOVE 1
TO
AREA-C
R-X

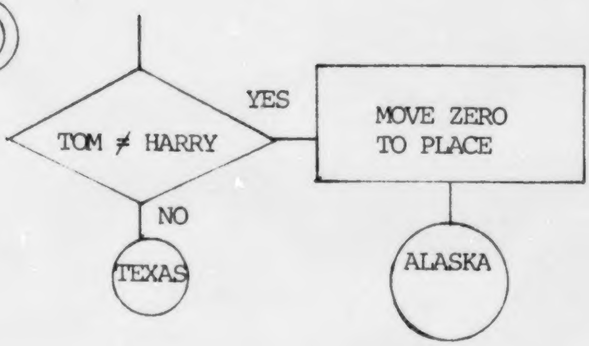
②


IS
CHUCK
NUMERIC
YES
TR
NO
CP

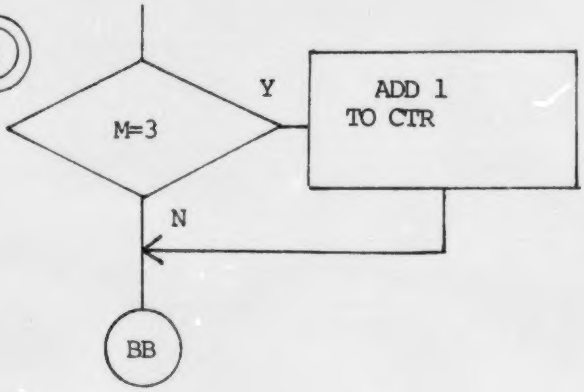
③


A=B
YES
STEP
3
NO
ADD 1
TO A

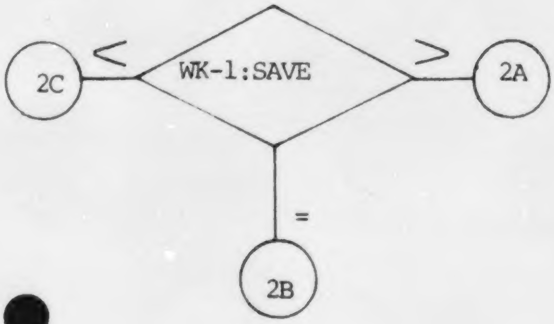
④


TOM ≠ HARRY
YES
MOVE ZERO
TO PLACE
NO
TEXAS
ALASKA

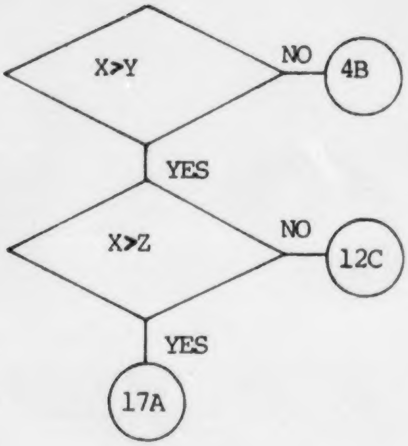
⑤

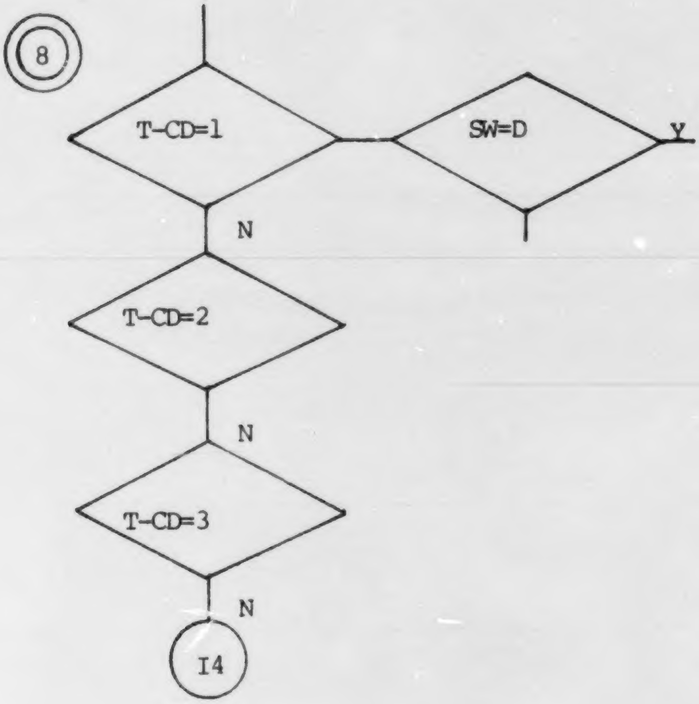

M=3
Y
ADD 1
TO CTR
N
BB

⑥


<
WK-1:SAVE
>
2C
2A
=
2B

⑦


X>Y
NO
4B
YES
X>Z
NO
12C
YES
17A


8
T-CD=1
SW=D
Y
N
T-CD=2
N
T-CD=3
N
I4

HANDOUT 03/04/03

PROBLEM STATEMENT 1

CREDIT UNION ACCOUNT

An employee deposited $650.00 at his credit union. The credit union votes an annual dividend of 3 1/2% interest. If he did not add any further to his savings, how much would he have at the end of 10 years, assuming a regular 3 1/2% dividend compounded annually?

Write a program which will print the value of the account and the number of years remaining each year.

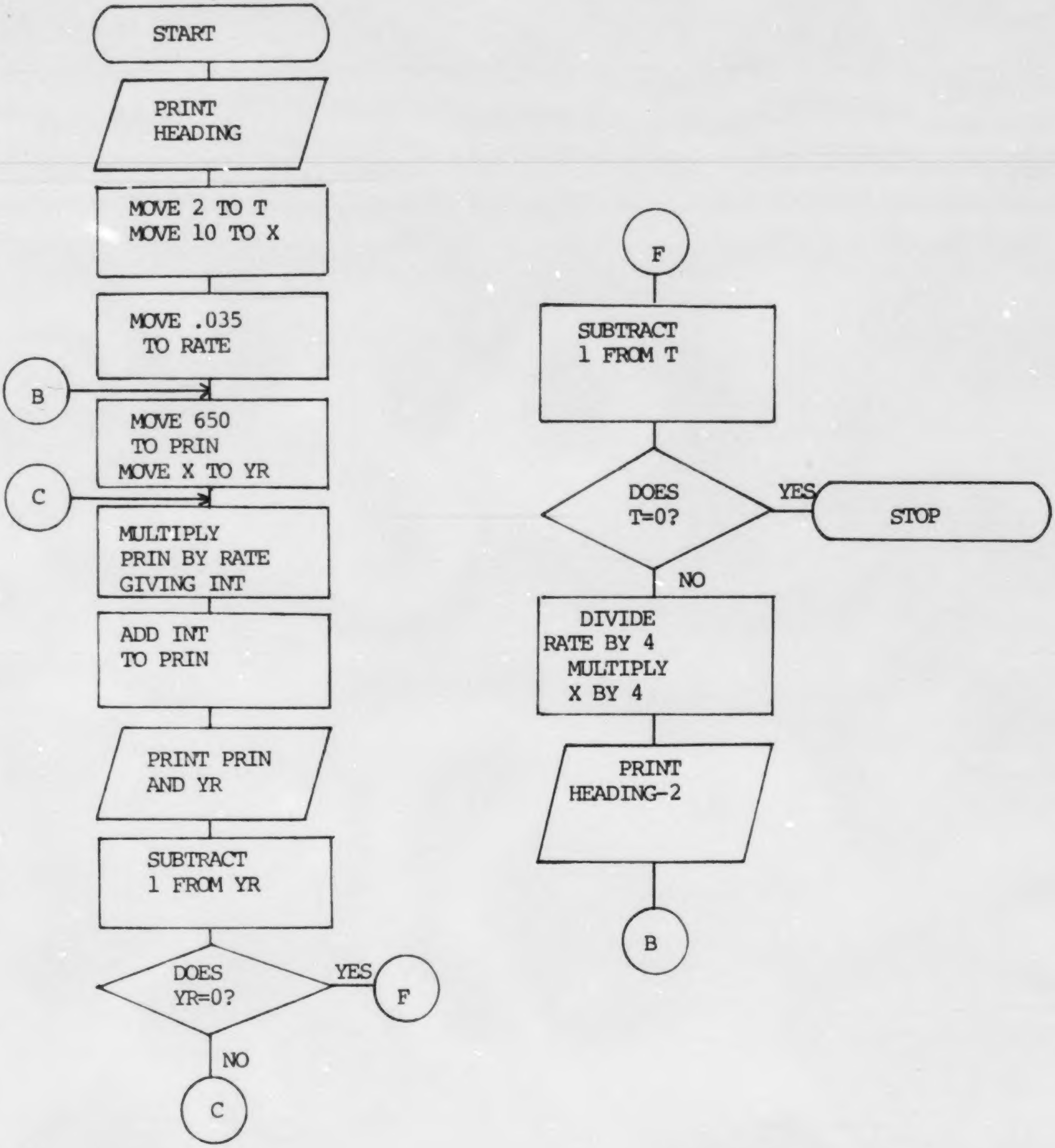
When the 10th year is printed, refigure the same principal and interest if the interest is compounded 4 times a year, and print a listing for each of the 40 quarters.

Notice that there is no input for this problem.

If you want an extra challenge, compute the interest quarterly, but only print it once for each year. This will require adding at least one more decision to the flowchart.

CREDIT UNION ACCOUNT - continued

FLOWCHART


START
PRINT HEADING
MOVE 2 TO T
MOVE 10 TO X
MOVE .035 TO RATE
B
MOVE 650 TO PRIN
MOVE X TO YR
C
MULTIPLY PRIN BY RATE GIVING INT
ADD INT TO PRIN
PRINT PRIN AND YR
SUBTRACT 1 FROM YR
DOES YR=0?
YES
F
NO
C
F
SUBTRACT 1 FROM T
DOES T=0?
YES
STOP
NO
DIVIDE RATE BY 4
MULTIPLY X BY 4
PRINT HEADING-2
B

PROBLEM STATEMENT 2

GRADE AVERAGE

Each student in the math class is to be listed showing the average of all his/her scores. Some students have more scores than others, so each one's average is based upon his/her own number of scores. The input is on cards; the output is a printed report.

Input:

1-15	NAME
16-18	GR-1
19-21	BLANK
22-24	GR-2
25-27	BLANK
28-30	GR-3
31-33	BLANK
34-36	GR-4
37-39	BALNK
40-42	GR-5
43-45	BLANK
46-48	GR-6
49-51	BLANK
52-54	GR-7
55-57	BLANK
58-60	GR-8
61-63	BLANK
64-66	GR-9
67-69	BLANK
70-72	GR-10
73-75	BLANK
76-78	GR-11
79-80	BLANK

Output:

1-20	BLANK
21-36	NAME
37-44	BLANK
45-47	AVERAGE (ROUNDED)
48-53	BLANK
54-55	NO-OF-SCORES (CTR)
56-132	BLANK

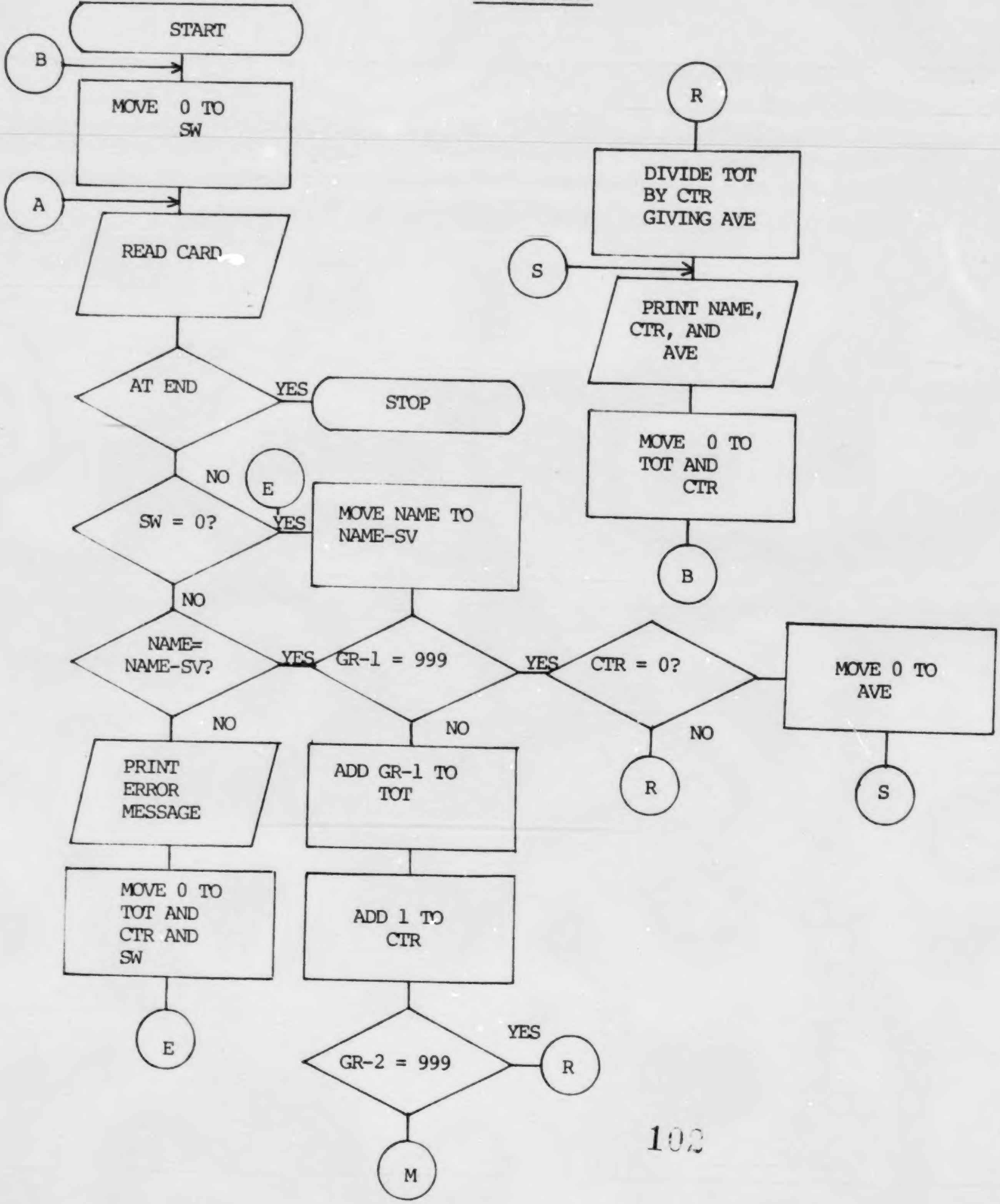
NOTE: The only difficulty in solving this problem is the matter of differing numbers of scores. To indicate the end of scores for a student, 999 is punched into the grade field following the last score to be included. The nines are an indicator only and are not to be used in any calculations. Example: If a student has more than 10 scores there will be another card with his/her name and the rest of his grade information. In this case there will be no nines on the first card. If there is a student with no scores, steps must be taken to avoid dividing by zero which will stop the program.

John has 5 grades (scores) to be averaged which, when added together and divided by 5, give him an average if 87.2, rounded off to 87.

JOHN DOE 085 077 091 096 087 999

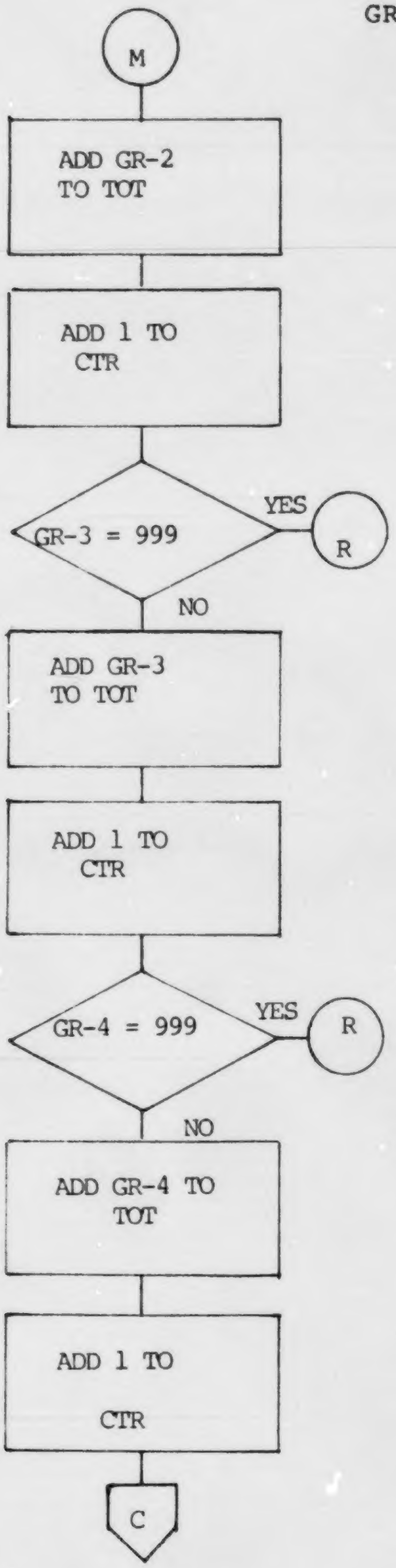
GRADE AVERAGE - continued

FLOWCHART


START
B
MOVE 0 TO SW
A
READ CARD
AT END
YES
STOP
NO
SW = 0?
E
YES
MOVE NAME TO NAME-SV
NO
NAME= NAME-SV?
YES
GR-1 = 999
YES
CTR = 0?
MOVE 0 TO AVE
NO
NO
NO
PRINT ERROR MESSAGE
ADD GR-1 TO TOT
R
S
MOVE 0 TO TOT AND CTR AND SW
ADD 1 TO CTR
E
YES
GR-2 = 999
R
M
R
DIVIDE TOT BY CTR GIVING AVE
S
PRINT NAME, CTR, AND AVE
MOVE 0 TO TOT AND CTR
B

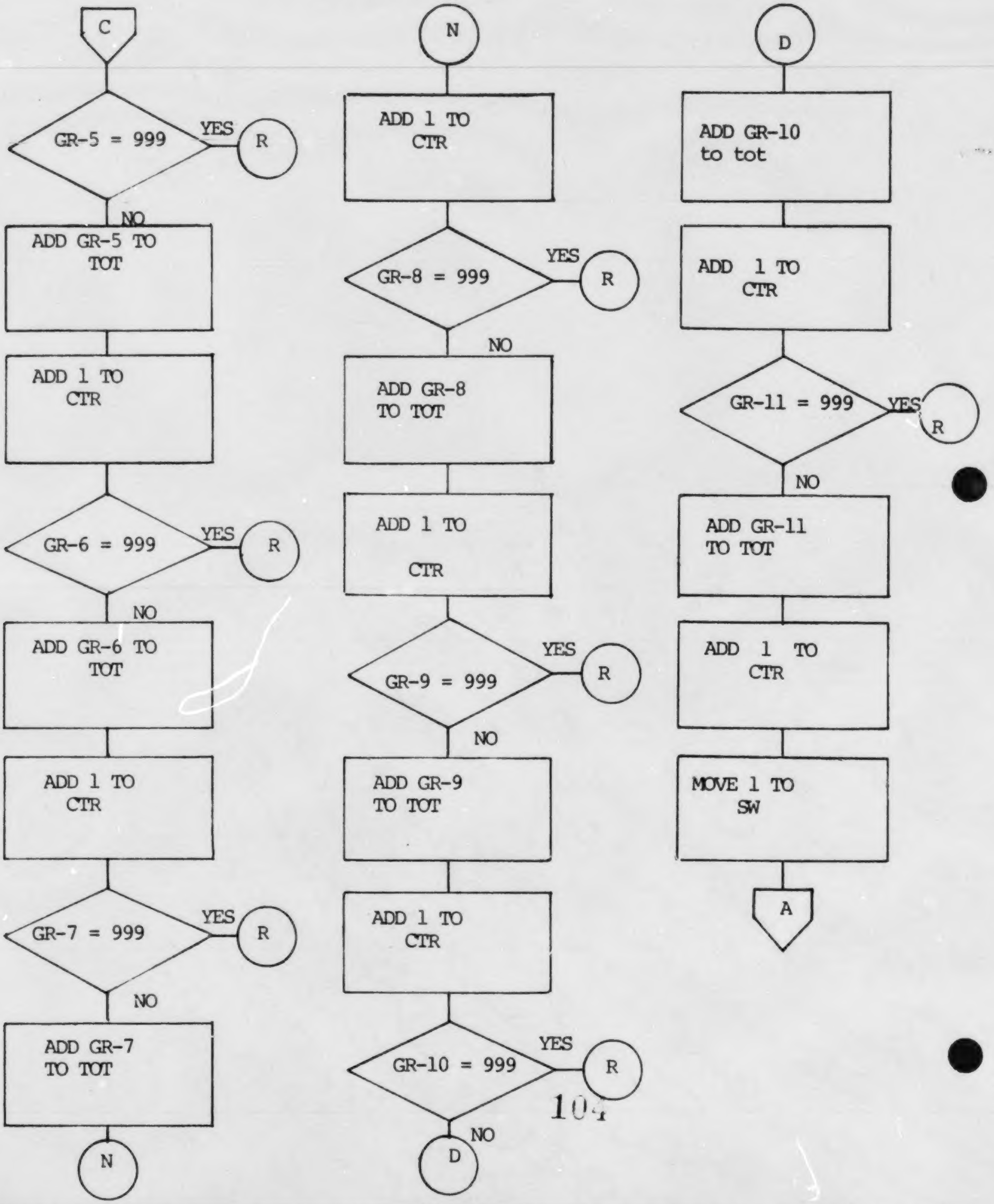
GRADE AVERAGE - continued

FLOWCHART - page 2


M
ADD GR-2 TO TOT
ADD 1 TO CTR
GR-3 = 999
YES
R
NO
ADD GR-3 TO TOT
ADD 1 TO CTR
GR-4 = 999
YES
R
NO
ADD GR-4 TO TOT
ADD 1 TO CTR
C

GRADE AVERAGE - continued

FLOWCHART - page 3


C
GR-5 = 999
YES
R
NO
ADD GR-5 TO TOT
ADD 1 TO CTR
GR-6 = 999
YES
R
NO
ADD GR-6 TO TOT
ADD 1 TO CTR
GR-7 = 999
YES
R
NO
ADD GR-7 TO TOT
N
N
ADD 1 TO CTR
GR-8 = 999
YES
R
NO
ADD GR-8 TO TOT
ADD 1 TO CTR
GR-9 = 999
YES
R
NO
ADD GR-9 TO TOT
ADD 1 TO CTR
GR-10 = 999
YES
R
NO
D
D
ADD GR-10 to tot
ADD 1 TO CTR
GR-11 = 999
YES
R
NO
ADD GR-11 TO TOT
ADD 1 TO CTR
MOVE 1 TO SW
A

EVALUATION 03/04

EMPLOYEE SUMMARY

The payroll department of a large manufacturing company requests that its data processing division supply its administrative department with a printed record containing certain information regarding all active employees. These records are on an 80-column card for each employee. The arrangement of the information within the columns is shown below:

Columns 1-20	Name of Employee
Columns 21-30	Blank
Column 31	A, B, or C (Occupation Code)
Columns 32-40	Blank
Columns 41-42	Number of years with Company
Columns 43-60	Blank
Columns 61-65	Five Digits-- Yearly Salary in Whole Dollars
Columns 66-73	Blank
Columns 74-78	Deductions in Whole Dollars
Column 79	Blank
Column 80	A = Active R = Retired

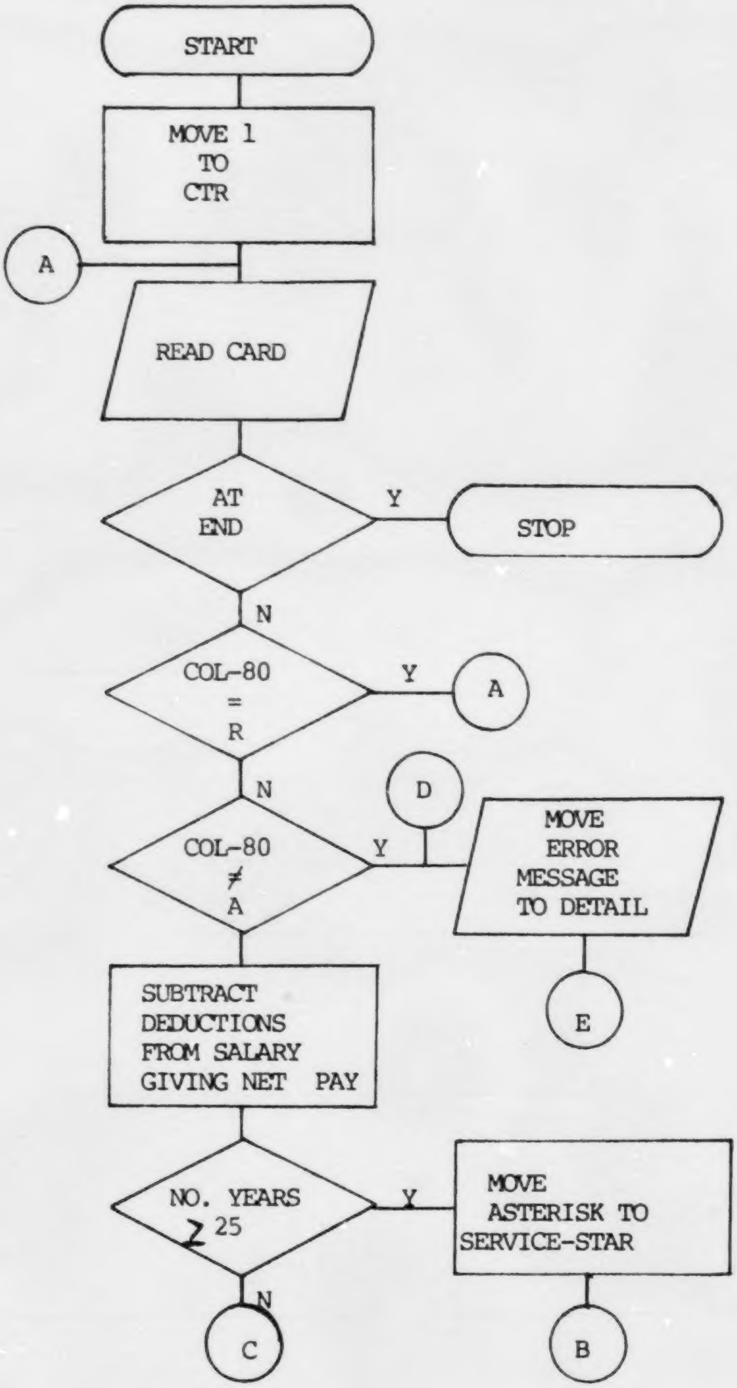
Columns 61 through 65 contain the yearly salary of the employee. Columns 74 through 78 contain the deductions made by the company from the employee's salary; they are stated on a full-year basis. The printed report will contain a new pay figure for each active employee. The printout which will be supplied to the company officials will either contain an "A" or an "R" in column 80. Any other character, including a blank (no punch), will be cause to write an error message. An "A" indicates that the card contains the record of an active employee and so should be processed. An "R" indicates that no processing is to occur for that particular record, since the output will list active employees only. Special note must be given to active employees with over 25 years service. These will be flagged by placing an asterisk on the line printed for any 25-year employee. Number of years in the employ of the company is indicated in columns 41 to 42.

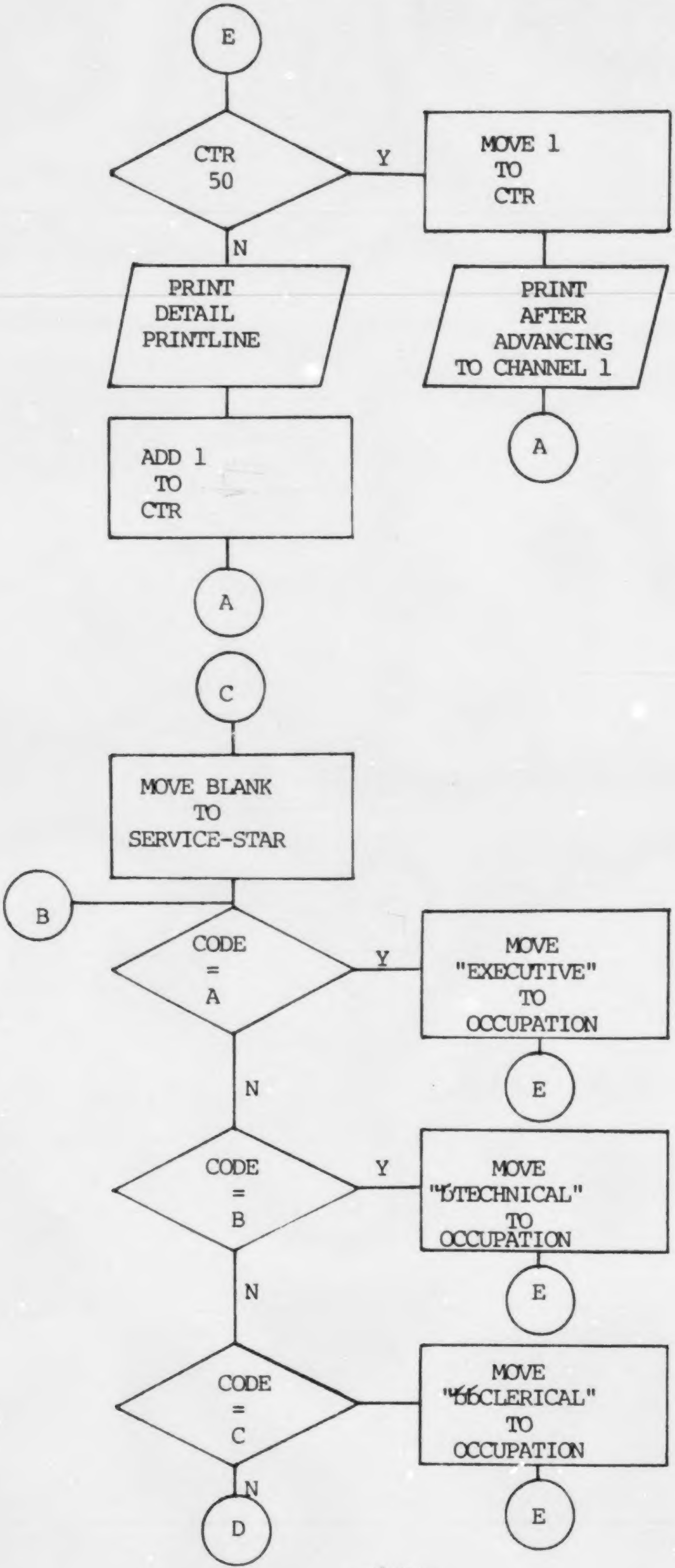
There are 132 print positions to a line and 50 employee lines on a page. After the information on the 50th employee has been printed, there will be a skip to the first line on the next page. The format of the printout is as follows:

Detail print line

1-10	Blank
11-30	Name
31-40	Blank
41-45	Net pay
46-54	Blank
55-64	Occupation
65-70	Blank
71	Service-star

FLOWCHART


START
MOVE 1 TO CTR
A
READ CARD
AT END
Y
STOP
N
COL-80 = R
Y
A
N
COL-80 ≠ A
Y
D
MOVE ERROR MESSAGE TO DETAIL
E
SUBTRACT DEDUCTIONS FROM SALARY GIVING NET PAY
NO. YEARS ≥ 25
Y
MOVE ASTERISK TO SERVICE-STAR
N
C
B


E
CTR
50
Y
N
MOVE 1
TO
CTR
PRINT
DETAIL
PRINTLINE
PRINT
AFTER
ADVANCING
TO CHANNEL 1
A
ADD 1
TO
CTR
A
C
MOVE BLANK
TO
SERVICE-STAR
B
CODE
=
A
Y
MOVE
"EXECUTIVE"
TO
OCCUPATION
E
N
CODE
=
B
Y
MOVE
"bTECHNICAL"
TO
OCCUPATION
E
N
CODE
=
C
MOVE
"bbCLERICAL"
TO
OCCUPATION
N
D
E

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 05: Use the editing features of COBOL to improve the readability of output.

PERFORMANCE GUIDE:

1. From flowcharts and problem statements, list data names for input, output, and work areas.
2. Develop and write data descriptions.
3. Finalize flowcharts and complete layouts and other documentation.
4. Prepare test data.
5. Code program on coding sheets.
6. Check coding for omissions and inconsistencies.
7. Enter program codes in preparation for compilation.
8. Proofread codes entered.
9. Enter job control codes in preparation for compilation.
10. Submit program for run.
11. Debug program using test data.

LEARNING ACTIVITIES

1. Read Chapters 28-31 in COBOL Simplified. Answer the questions at the end of each chapter and have your answers checked.

2. Study the sections in your manufacturer's COBOL reference manual on PICTURE, EDITING, and JUSTIFIED.

3. Answer these questions:

   a. What is meant by a "floating dollar sign," and when might you want one?
   b. What else can "float" besides a dollar sign?
   c. What does "zero suppression" mean?
   d. Why is it frequently desirable to suppress zeros?
   e. How is check protection related to editing in COBOL?

4. Prepare and compile the programs for the two problem statements in Handout 03/05/04. Use the procedures listed above in the Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, equipment reference manuals, data entry equipment, and coding sheets, prepare the required COBOL program and obtain an error-free compile within five computer runs.

HANDOUT 03/05/04

PROBLEM STATEMENT 1

DISCOUNT LIST

The XYZ Sales Company wants to print a list each day showing the amount of discounts allowed for prompt payment. The amounts for which discounts are to be calculated are punched into cards. These discounts are: 3% for payment within 10 days; 2% for payment within 20 days; and 1% for payment within 30 days.

Your listing should be well spaced and easy to read. Print dollar signs and decimal points. Suppress non-significant zeros to the left of the decimal point. Do not float the dollar sign.

Draw a layout with the title "DISCOUNT LIST" centered at the top. Use column headings as follows: "GROSS PAYMENT," "BILLING DATE," "PAYMENT DATE," "DISCOUNT," and "NET PAYMENT." Data items should be centered below their column headings and should be from 1 to 1-½ inches apart.

Input:

BILLING DATE, MONTH (1-2) DAY (3-4) YR. (5-6)
GROSS AMOUNT (7-11) 999V99 (V represents an assumed decimal point.)

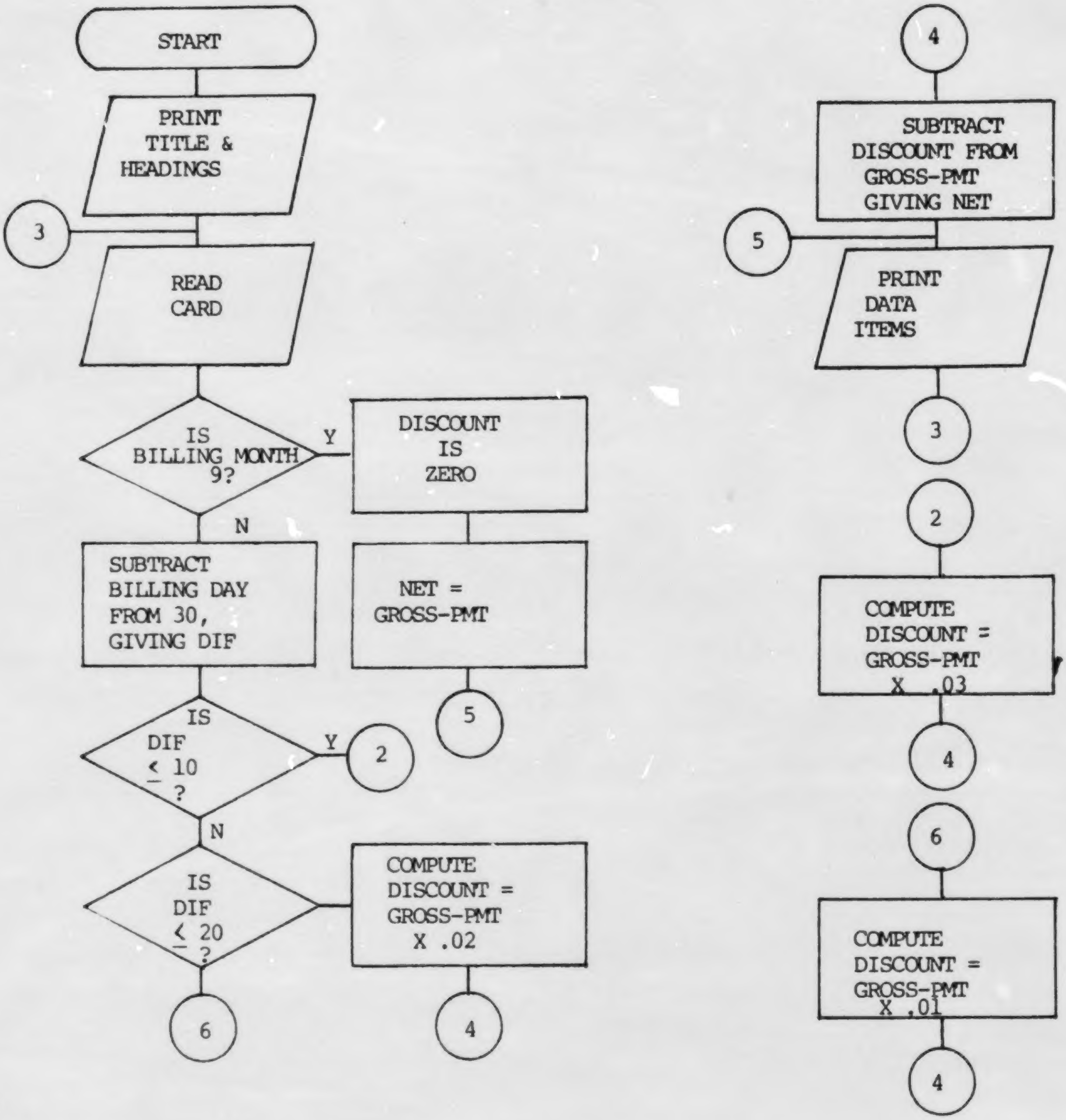
Bonus: Keep up with totals of gross payments, all discounts, and net payments; print these at the end of the report.

Extra Bonus: Notice that the flowchart is written for the date September 30. For most other dates the time calculation is more difficult. Try flowcharting the logic that would work for any month or day.

03/05/04

DISCOUNT LIST - continued

FLOWCHART


START
PRINT TITLE & HEADINGS
3
READ CARD
IS BILLING MONTH 9?
Y
DISCOUNT IS ZERO
NET = GROSS-PMT
5
N
SUBTRACT BILLING DAY FROM 30, GIVING DIF
IS DIF ≤ 10 ?
Y
2
N
IS DIF ≤ 20 ?
COMPUTE DISCOUNT = GROSS-PMT X .02
4
6
4
SUBTRACT DISCOUNT FROM GROSS-PMT GIVING NET
5
PRINT DATA ITEMS
3
2
COMPUTE DISCOUNT = GROSS-PMT X .03
4
6
COMPUTE DISCOUNT = GROSS-PMT X .01
4

03/05/04

PROBLEM STATEMENT 2

COMMISSION CHECK

The printing of checks presents a challenge to the program coder because the spacing of the output is so important. Each item must fit precisely in the allotted space on the printed form. You are to use the attached replica of a check when designing your print layout, and, of course, the check protection feature of COBOL is to be used.

Input:
- DATA-CARD
  - NAME 1-23
  - SALES 73-79 (99999V99)
  - COMMISSION RATE 33-35 (V999)
  - BONUS 36-40 (999V99)
  - AMOUNT DRAWN 41-46 (9999V99)
  - TAX DEDUCTIONS 47-51 (999V99)
  - SS # 24-32
- DATE CARD
  - DATE 1-6
  - CODE "1" 80

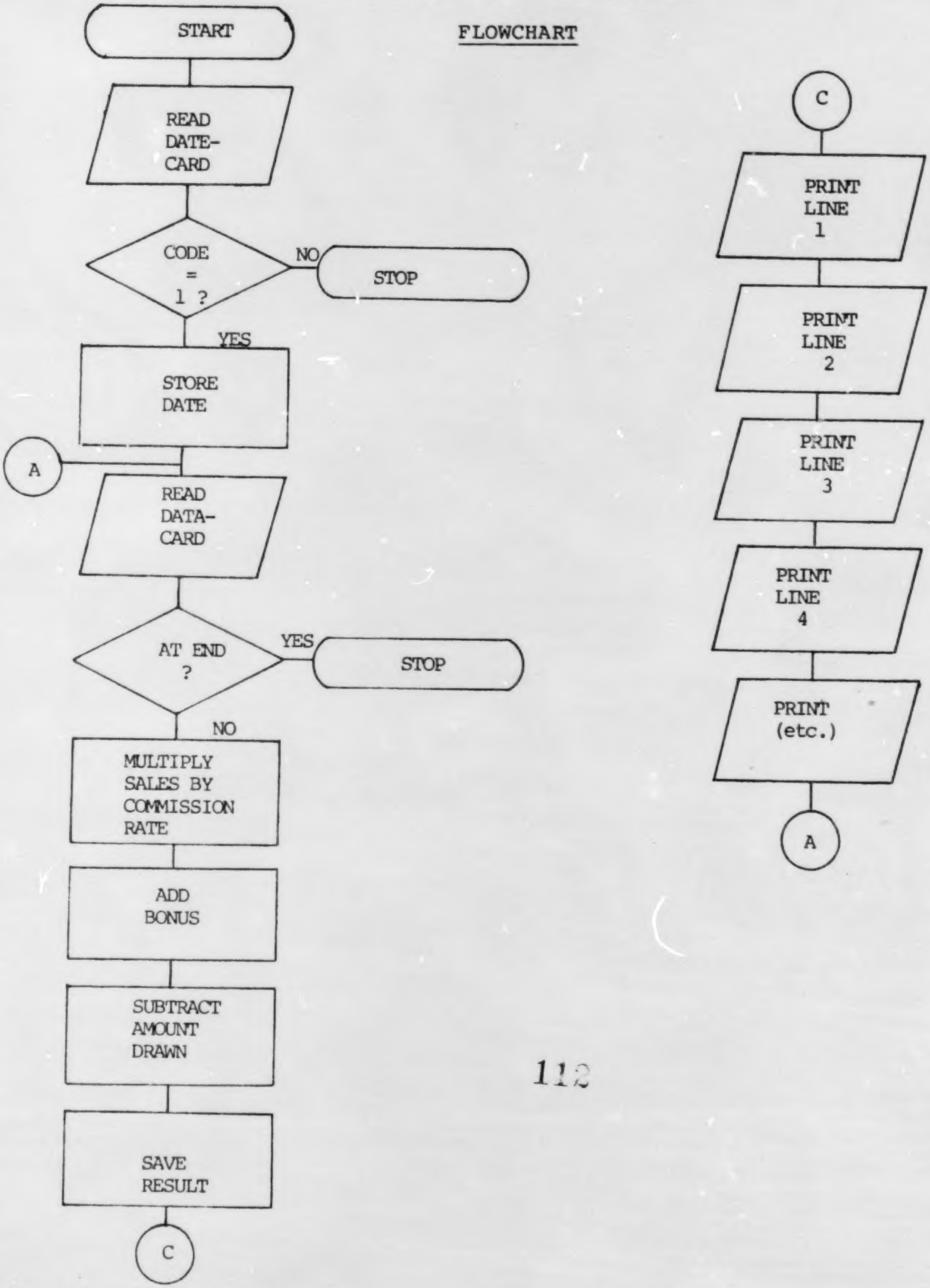
Output:
- CHECK
  - CURRENT DATE
  - CHECK NUMBER
  - PAYEE
  - AMOUNT (2 places)
  - SS #
- VOUCHER
  - DATE NAME
  - CHECK NUMBER
  - GROSS COMMISSION
  - AMOUNT DRAWN
  - DEDUCTIONS
  - CHECK AMOUNT
  - SS #

The current date is to be entered by use of a "date card," which is the first card in the data deck.

The check number is originated as part of the program. The first check this run is number 109046, and the remaining checks are to be numbered in ascending sequence following this. The date should be printed in both places with slashes between month, day, and year.

COMMISSION CHECK - continued

FLOWCHART


START
READ DATE-CARD
CODE = 1 ?
NO
STOP
YES
STORE DATE
A
READ DATA-CARD
AT END ?
YES
STOP
NO
MULTIPLY SALES BY COMMISSION RATE
ADD BONUS
SUBTRACT AMOUNT DRAWN
SAVE RESULT
C
C
PRINT LINE 1
PRINT LINE 2
PRINT LINE 3
PRINT LINE 4
PRINT (etc.)
A

03/05/04

COMMISSION CHECK

GIANT SALES CORP.

Check No. ______

Date __________

Pay to the order of ____________________________ Amt. __________

____ DOLLARS

____

PAYROLL ACCOUNT

CHECK NO. ☐

DATE ☐

NAME ☐ SS# ☐

GROSS ☐ AMT. DRAWN ☐

DEDUCTIONS ☐ AMT. PAID ☐

GIANT SALES CORP.

EVALUATION 03/05

CHARGE ACCOUNT STATEMENT

Write a program to print the monthly statements to charge account customers at the Long Department Store.

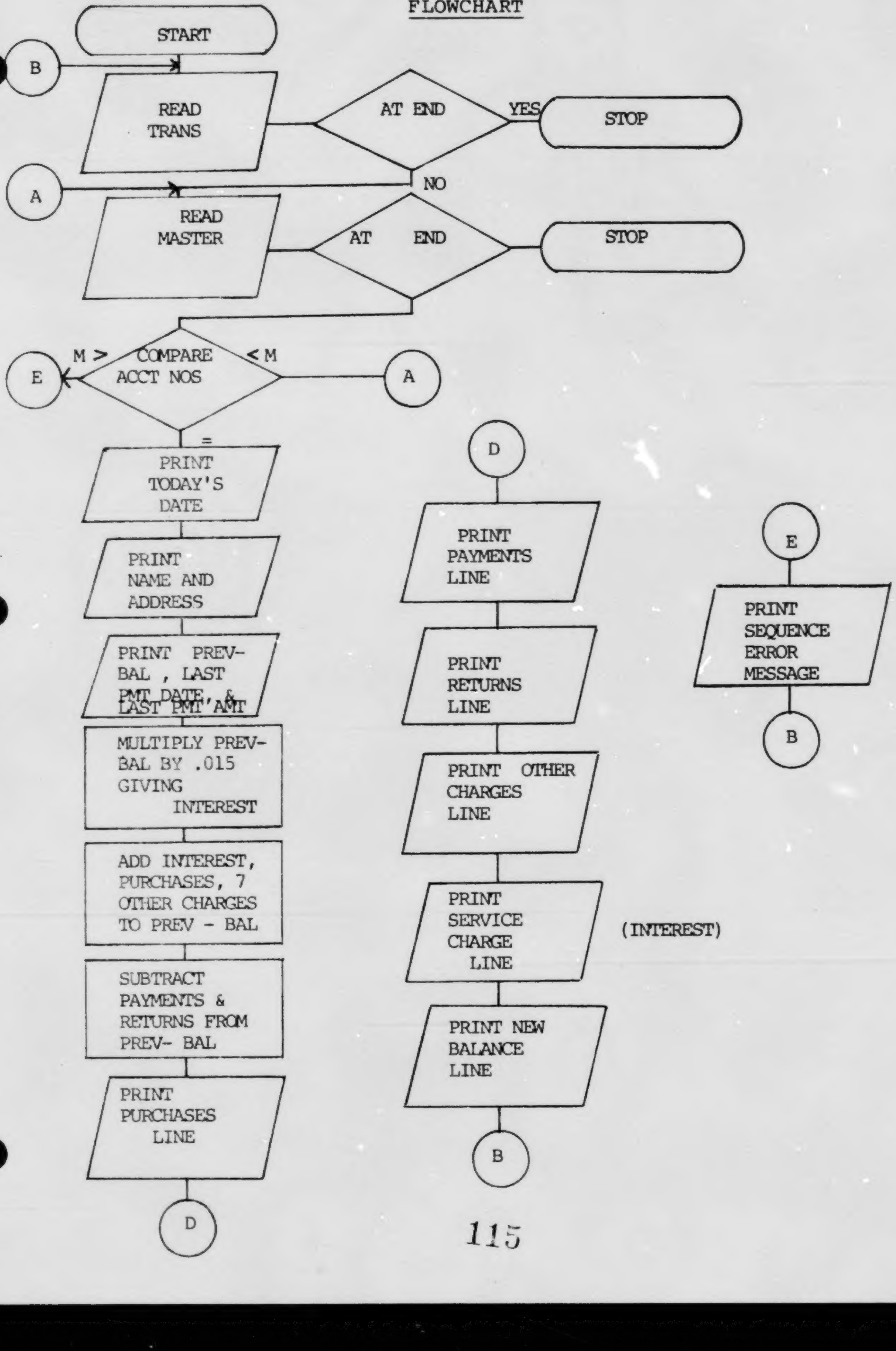
Printing is to be done on a special form and each item must fit exactly in its assigned space. The statement is to be folded in half and inserted in a window envelope with the name and address showing.

Input:
- MASTER FILE TAPE
  - NAME (26)
  - ADDRESS (26)
  - CITY (15)
  - STATE (5)
  - ZIP (5)
  - BALANCE (6) 9999V99
  - ACCOUNT NO. (5)
  - CREDIT LIMIT (4) 999
  - BILLING DATE (DD) (2)
  - LAST PMT. AMT. (6) 9999V99
  - LAST PAYMENT DATE (MMDDYY)

  - TRANSACTION FILE CARDS
  - ACCOUNT NO. (5)
  - PURCHASES (DEBITS) (6) 9999V99
  - PAYMENTS (CREDITS) (6) 9999V99
  - RETURNS (") (5) 999V99
  - OTHER CHARGES (CREDITS) (5) 999V99

Output: (see sample form)
- DATE
- NAME
- ADDRESS
- CITY, STATE, ZIP
- PREVIOUS BALANCE
- PURCHASES
- RETURNS
- PAYMENTS
- OTHER CHARGES
- SERVICE CHARGE
- LAST PMT. DATE
- LAST PMT. AMOUNT
- ACCOUNT NO.
- BALANCE

FLOWCHART


START
B
READ TRANS
AT END
YES
STOP
NO
A
READ MASTER
AT END
STOP
M >
COMPARE ACCT NOS
< M
E
A
=
PRINT TODAY'S DATE
PRINT NAME AND ADDRESS
PRINT PREV-BAL , LAST PMT DATE, & LAST PMT AMT
MULTIPLY PREV-BAL BY .015 GIVING INTEREST
ADD INTEREST, PURCHASES, 7 OTHER CHARGES TO PREV - BAL
SUBTRACT PAYMENTS & RETURNS FROM PREV- BAL
PRINT PURCHASES LINE
D
D
PRINT PAYMENTS LINE
PRINT RETURNS LINE
PRINT OTHER CHARGES LINE
PRINT SERVICE CHARGE LINE
(INTEREST)
PRINT NEW BALANCE LINE
B
E
PRINT SEQUENCE ERROR MESSAGE
B

EVALUATION 03/05

LONG DEPARTMENT STORE

DATE ☐

NAME
ADDRESS
CITY, STATE, ZIP

BALANCE ☐

PREVIOUS BALANCE ☐
PURCHASES ☐
RETURNS ☐
PAYMENTS ☐
OTHER CHARGES ☐
SERVICE CHARGES ☐
(This means interest)

These should match

LAST PAYMENT DATE ☐ AMOUNT (of last pmt.) ☐

ACCOUNT NO. ☐ BALANCE ☐

STUDENT COMPETENCY SHEET

MODULE 03: COBOL Programming

COMPETENCY 06: Manipulate data in table form using COBOL program features.

PERFORMANCE GUIDE:

1. From flowcharts and problem statements, list data names for input, output, and work areas.
2. Develop and write data descriptions.
3. Finalize flowcharts and complete other documentation.
4. Prepare table layout and other layout specifications.
5. Prepare test data.
6. Code program on coding sheets.
7. Check coding for omissions and inconsistencies.
8. Enter program codes in preparation for compilation.
9. Proofread codes entered.
10. Enter job control codes in preparation for compilation.
11. Submit program for run.
12. Debug program using diagnostics.
13. Test program using test data.

LEARNING ACTIVITIES

1. Read Chapters 37-39 in COBOL Simplified. Answer the questions at the end of each chapter and have your answers checked.

2. Study the sections in your manufacturer's COBOL reference manual on Tables, Subscripting, PERFORM, and OCCURS.

3. Complete Handout 03/06/03. Have your answers checked.

4. Prepare and compile the programs for the two problem statements in Handout 03/06/04. Use the procedures listed above in the Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, equipment reference manual, data entry equipment, and coding sheets, prepare the required COBOL program and obtain an error-free compile within five computer runs.

HANDOUT 03/06/03

Instructions

When writing a program which contains a table (or tables), there are certain rules to remember.

A. The table item must be named and it must be described by a level greater than 01.

B. The computer must be told how much space to allow for the table by indicating:

1. the size (and data type) of each item, and
2. the number of items to be put in the table.

C. All items in the table must be of the same size and type.

D. When referring to a table item, a subscript in parenthesis must follow the name.

E. The subscript must have a numeric value greater than zero, but no more than the number of table items.

1. Write COBOL statements to store the following data in a table. Use an 01 level entry to name the table.
   SPANIEL, POODLE, GERMAN SHEPHERD, CHIHUAHUA, DACHSHUND,
   BEAGLE, COLLIE, BLOODHOUND, FOX TERRIER, DALMATIAN.

2. Write COBOL statements for this data using a table. The data is contained in punched cards. All information except code and date are to be placed in working storage.

   Card 1 - Clerk Name (1-20)
   Clerk Number (21-25)
   Commission Rate (26-29)
   Code (80)

   Cards 2, 3, 4, 5, 6, 7 - Date (1-6)
   Clerk Number (21-25)
   Daily Sales (30-35) 9999V99
   Code (80)

   Write FD entries, working storage entries, and the Procedure Division statements to read and store data.

3. Write all necessary COBOL entries to read the following card data

and place the rates in a table in storage. Include both Data Division and Procedure Division entries.

Merchandise Category (1-4)
Weight in pounds (5-7)
Postage Rates (8-11)
" " (12-15)
" " (16-19)
" " (20-23)
" " (24-47)
" " (28-31)
" " (32-35)
" " (36-39)

If you did not use a PERFORM statement in the last two problems, you did a lot of unnecessary work. Rewrite the Procedure entries using PERFORM.

4. Using the same data as problem 3, write statements, including subscript, that would cause a list of rates to be printed.

HANDOUT 03/06/04

PROBLEM STATEMENT 1

STATE GASOLINE TAX

Enter the name of each state along with its current gasoline tax into a table. This data is punched into cards, one state to a card. There are no more than 50 state cards, but there may be less. There will be one card following the state cards, which contains 999 in the tax field as a signal of the end.

Following this card, and having the same format, are cards which contain changed rates for some states. These rates are to replace the original ones in the table.

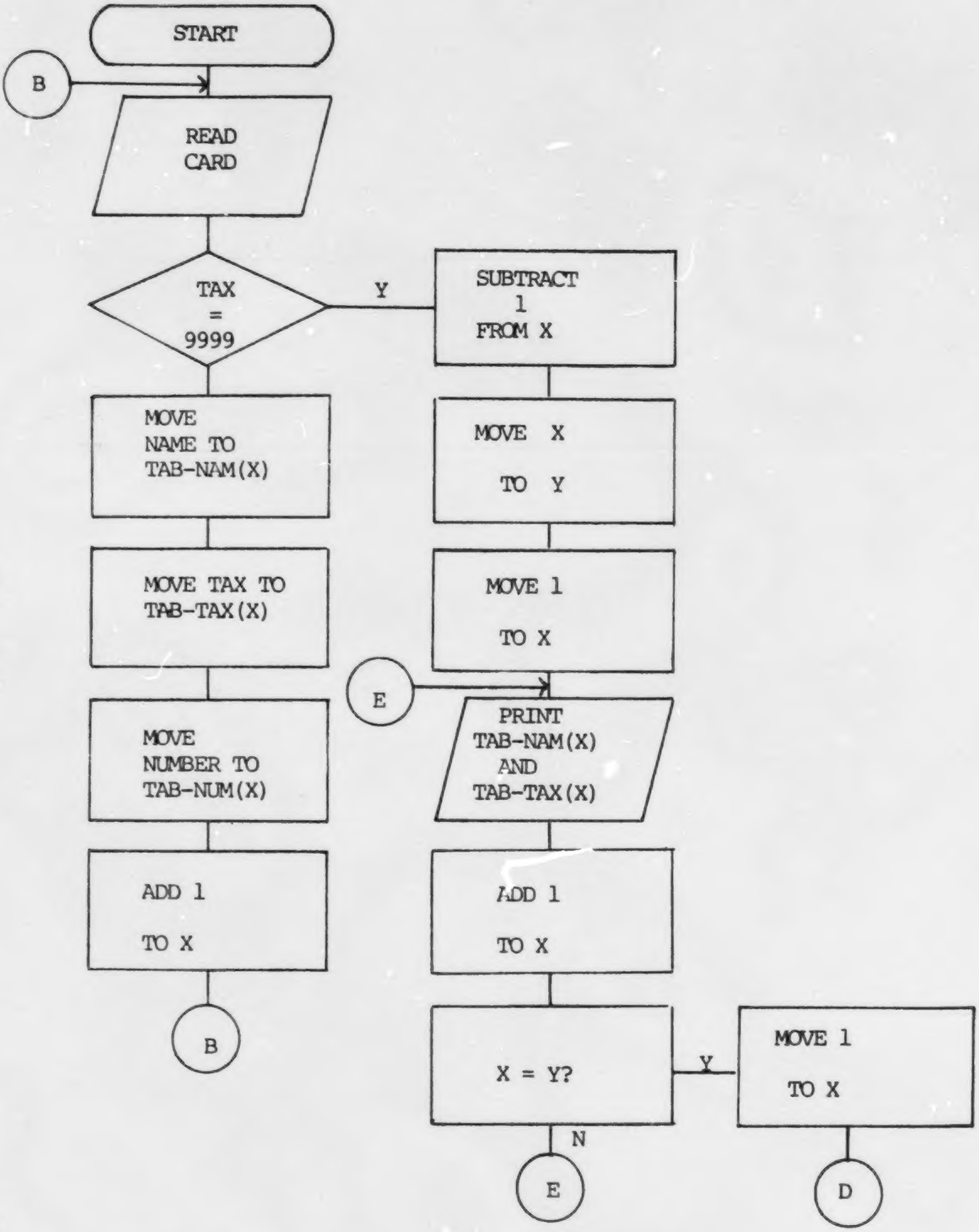
Write the program to read the data into a table, print a list, read the changes into the table, and print an updated list. Indicate a heading at the beginning of each list. Give x a value of 1 in the data division.

```
    Input:  Col.  1-14  STATE NAME
            Col.  15-18 GASOLINE TAX (V9999)
            Col.  79-80 NUMBER
```

03/06/04

STATE GASOLINE TAX - continued

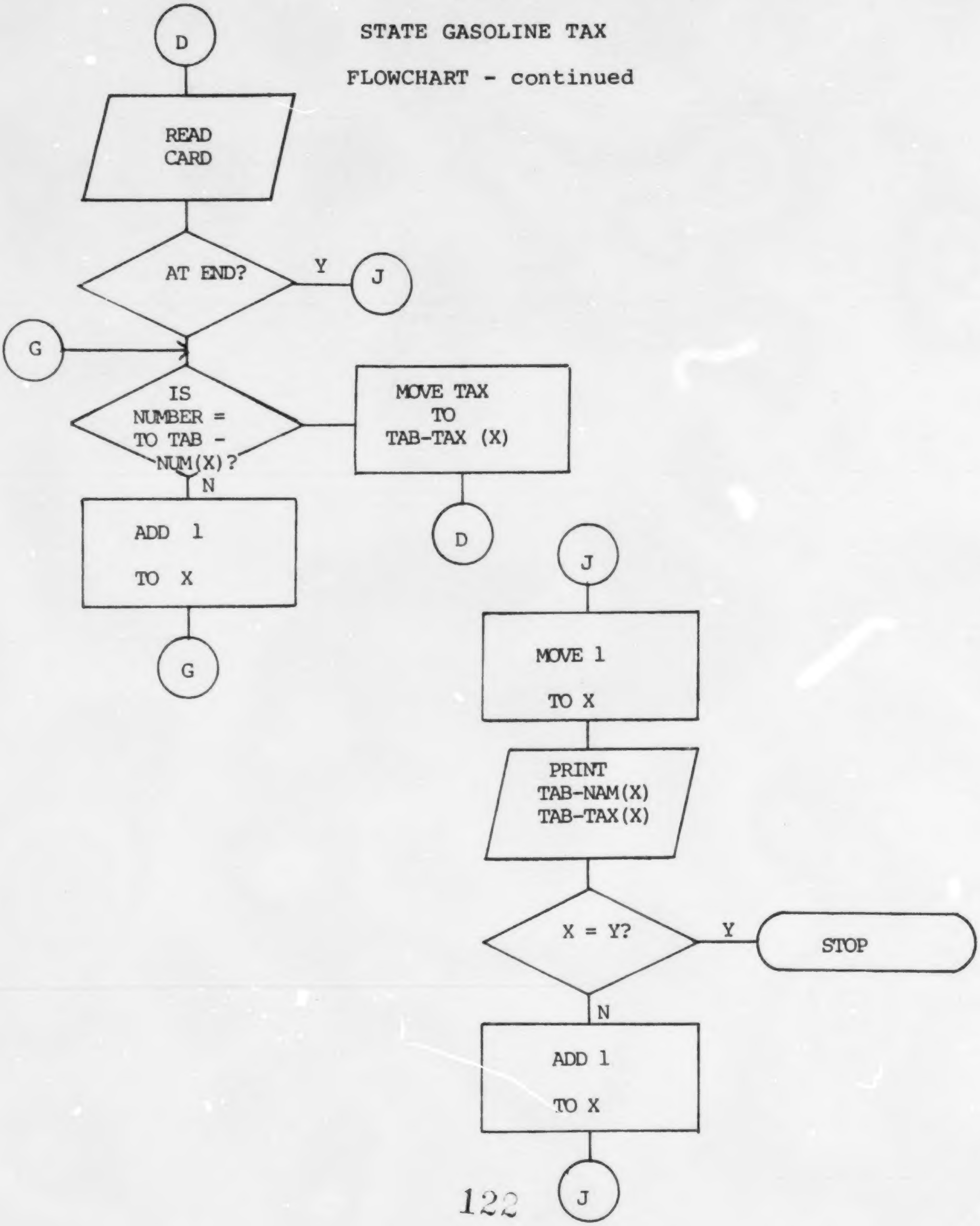
FLOWCHART


START
B
READ CARD
TAX = 9999
Y
SUBTRACT 1 FROM X
MOVE NAME TO TAB-NAM(X)
MOVE X TO Y
MOVE TAX TO TAB-TAX(X)
MOVE 1 TO X
E
PRINT TAB-NAM(X) AND TAB-TAX(X)
MOVE NUMBER TO TAB-NUM(X)
ADD 1 TO X
ADD 1 TO X
B
X = Y?
Y
MOVE 1 TO X
N
E
D

03/06/04

STATE GASOLINE TAX

FLOWCHART - continued


D
READ CARD
AT END?
Y
J
G
IS NUMBER = TO TAB - NUM(X)?
N
MOVE TAX TO TAB-TAX (X)
D
ADD 1 TO X
G
J
MOVE 1 TO X
PRINT TAB-NAM(X) TAB-TAX(X)
X = Y?
Y
STOP
N
ADD 1 TO X
J

03/06/04

PROBLEM STATEMENT 2

TEA MERCHANTS

This problem requires a few techniques which you may not have encountered before.

A. Your input file (card file) contains 2 record types, each of which is to be described under the FD.

B. The first record type contains an item (qty) which occurs 4 times on the card and which is to be described as a small table. Do the same thing for the 4 qty fields on the output detail line.

C. There are 3 dimensions to the main table, so all references to the individual items must list 3 subscripts.

D. Tea names are stored in a separate table of 10 items. City names are to be entered as literals in working storage as they are not included in the input; use a 3-item table.

E. Error routines must be included to avoid getting hung in a loop due to a data error, such as an incorrect code. There are numerous error possibilities which are not accounted for, but if the flowchart is followed, they would result in termination rather than endless looping. Connections are shown on the flowchart for several error checks, but routines are not shown. Write an error message for each condition indicated and terminate the program. One error check is to terminate the program if there are more or less than 30 master cards. After your program appears to be running correctly, alter the data deck to test your error-routines.

Initial values of counters, switches and subscripts are:

A = 10
Y = 1
Z = 0
C-CTR = o
SW = 1
X = 0

03/06/04

TEA MERCHANTS

3 DIMENSIONAL TABLE PROBLEM

1. The Hung Wun On Chai Merchants have trading compounds in Calcutta, Hong Kong, and Shanghai. They stock 10 kinds of tea, and in 4 qualities, namely, for Rajas, Nabobs, Sahibs, and Coolies.

2. There is a deck of 30 cards, with the following layout:

cc 2	City Code (C-CD)
cc 3 - 5	Tea Code Number (T-CD)
cc 6 - 25	Tea Name (NAME)
cc 31 - 50	4 Quantity on hand fields, each 5 columns long (QTY) (Subscripted)

3. First, devise and draw a picture of the 3-D Table, to make sure you understand the row, column, and rank involved. Get it checked.

   Second, divise the data structure to hold the table in working-storage.

4. Read the 30 cards and put the table into working-storage.

5. Following the Table Master cards are some transactions cards, coded as follows:

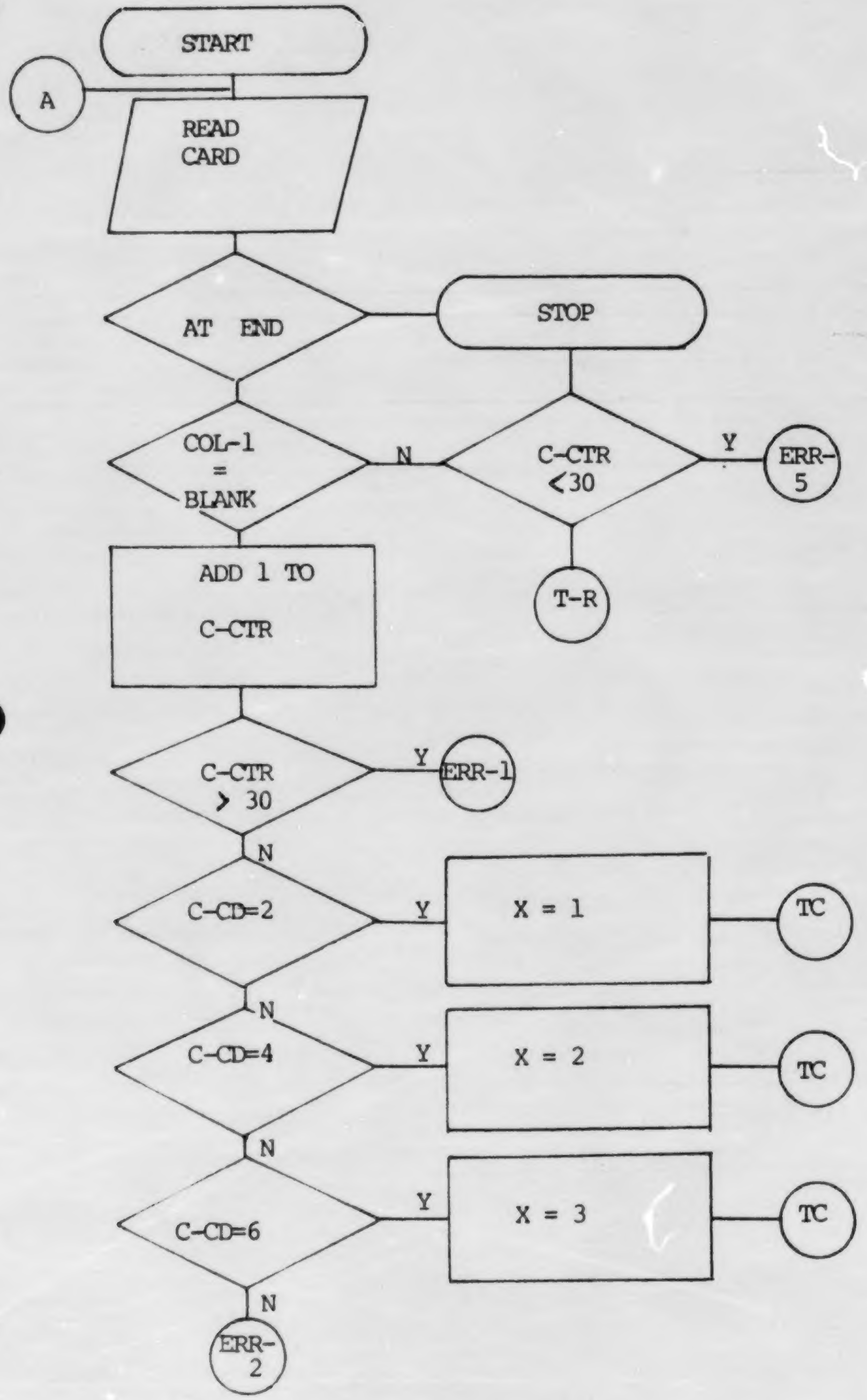
cc 1	City code (C-CD-1)
cc 2-3	Tea Number code (T-CD-1)

6. After building the table, read the transactions cards. Use their data to retrieve information from the table. Extract and print the following items, one line of print per card.

```
                      HUNG WUN ON CHAI MERCHANTS

                       HONG KONG   MALAY STATES

City Name        Tea Name                  Quantity on hand
                                  Raja    Nabob    Sahib    Coolie

XXXXXXXXX        XXXXXXXXXXX      XXXXX   XXXXX    XXXXX    XXXXX
 (C-OUT)           (N-OUT)                  (Q-OUT)
```

TEA MERCHANTS PROBLEM - continued

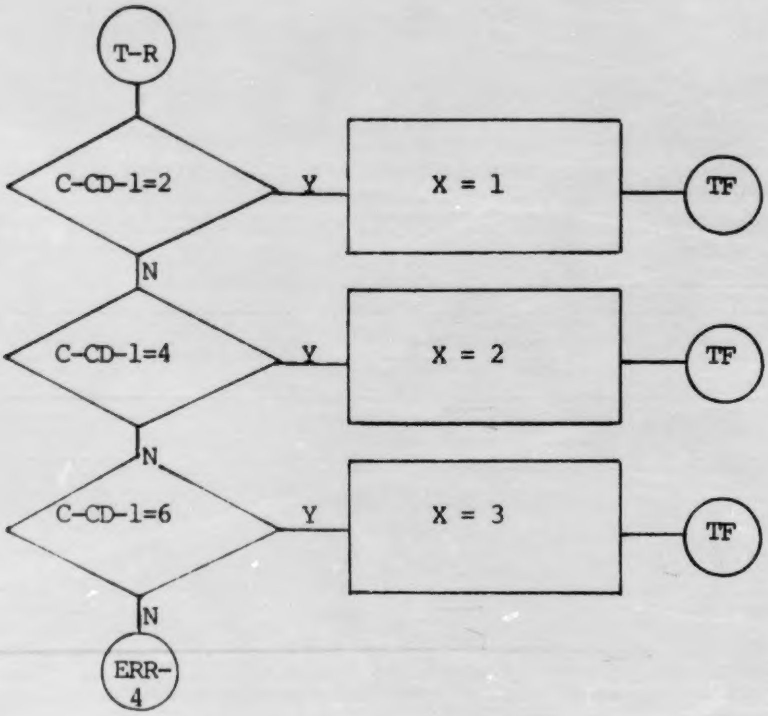
FLOWCHART - Pg. 1 of 3

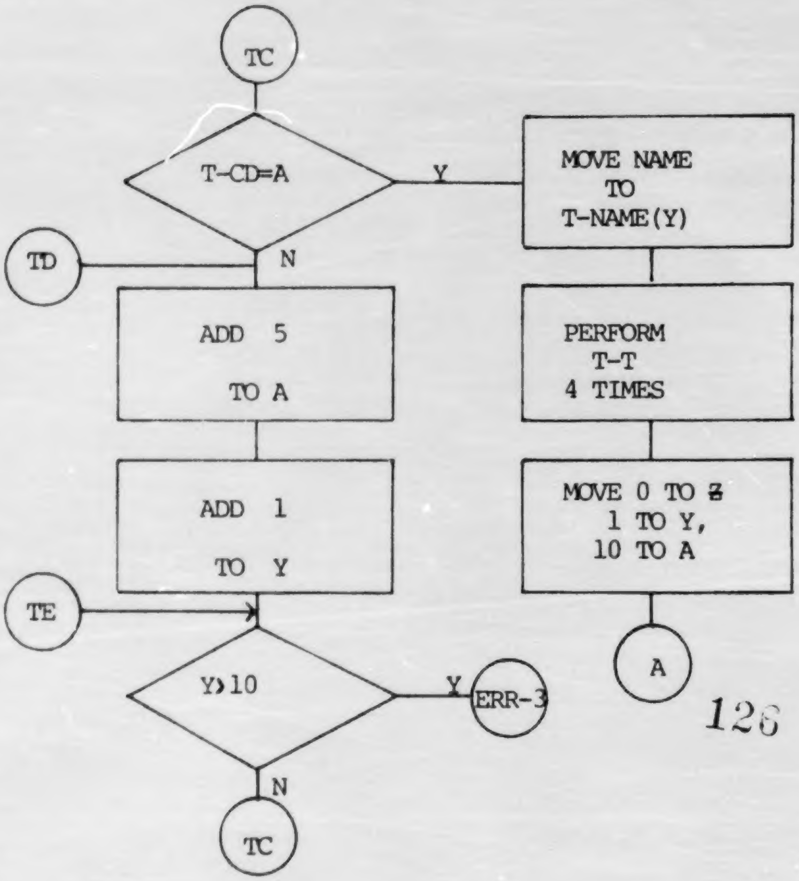

START
A
READ CARD
AT END
STOP
COL-1 = BLANK
N
C-CTR <30
Y
ERR-5
T-R
ADD 1 TO C-CTR
C-CTR > 30
Y
ERR-1
N
C-CD=2
Y
X = 1
TC
N
C-CD=4
Y
X = 2
TC
N
C-CD=6
Y
X = 3
TC
N
ERR-2

TEA MERCHANTS PROBLEM - continued

FLOWCHART - Pg. 2 of 3

03/06/04

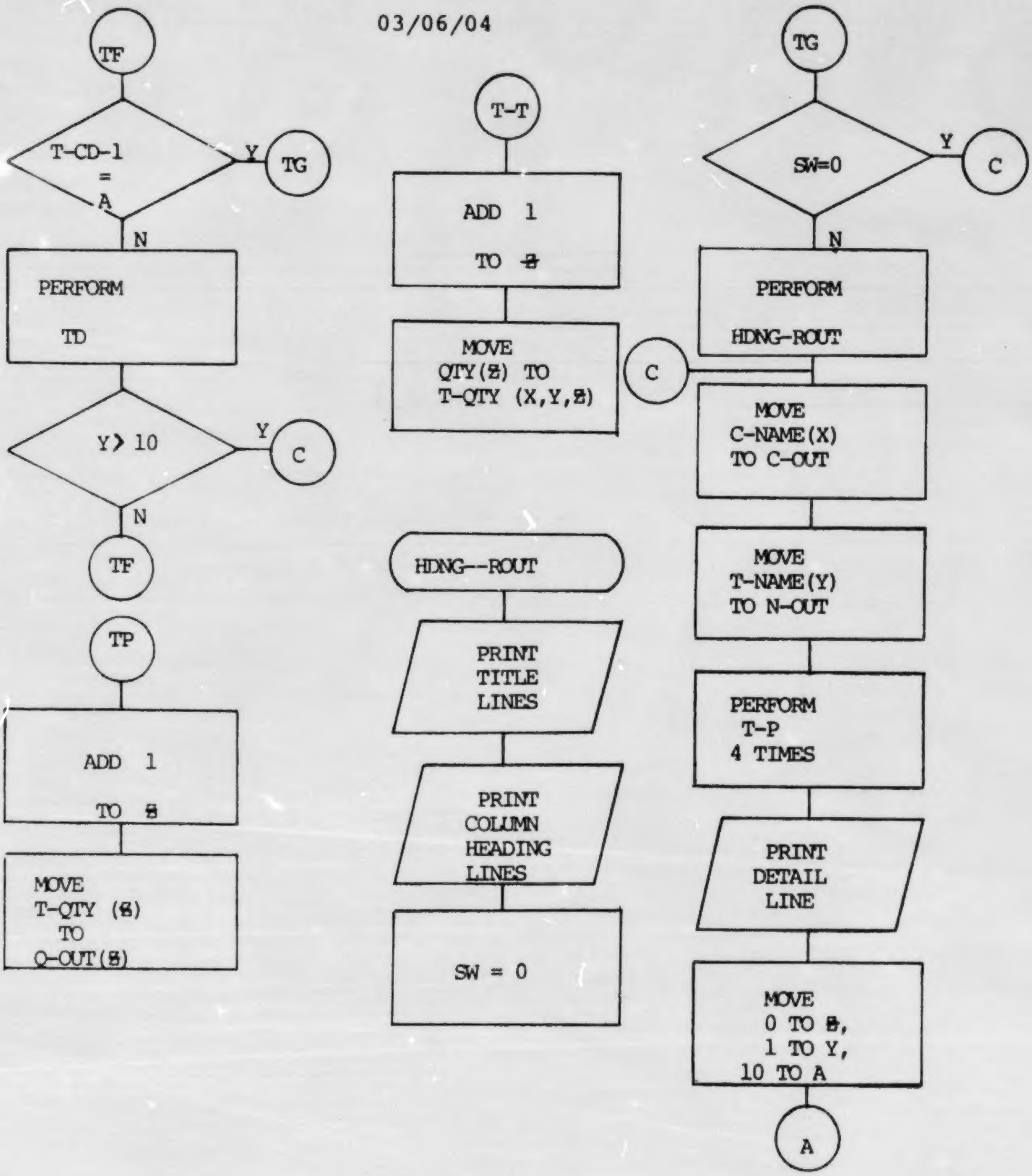

T-R
C-CD-1=2
Y
X = 1
TF
N
C-CD-1=4
Y
X = 2
TF
N
C-CD-1=6
Y
X = 3
TF
N
ERR-4


TC
T-CD=A
Y
MOVE NAME TO T-NAME(Y)
TD
N
ADD 5 TO A
PERFORM T-T 4 TIMES
ADD 1 TO Y
MOVE 0 TO Z 1 TO Y, 10 TO A
TE
A
Y>10
Y
ERR-3
N
TC

TEA MERCHANTS PROBLEM - continued

FLOWCHART - Pg. 3 of 3

03/06/04


TF
T-CD-1
=
A
Y
TG
N
PERFORM
TD
Y> 10
Y
C
N
TF
TP
ADD 1
TO Z
MOVE
T-QTY (Z)
TO
Q-OUT(Z)
T-T
ADD 1
TO Z
MOVE
QTY(Z) TO
T-QTY (X,Y,Z)
HDNG--ROUT
PRINT
TITLE
LINES
PRINT
COLUMN
HEADING
LINES
SW = 0
TG
SW=0
Y
C
N
PERFORM
HDNG-ROUT
C
MOVE
C-NAME(X)
TO C-OUT
MOVE
T-NAME(Y)
TO N-OUT
PERFORM
T-P
4 TIMES
PRINT
DETAIL
LINE
MOVE
0 TO Z,
1 TO Y,
10 TO A
A

EVALUATION 03/06

WITHHOLDING TAX CALCULATIONS

To calculate the withholding deductions of its salaried employees, Dingaling Enterprises uses the tables provided by the Internal Revenue Service. The necessary information for calculating total yearly taxes is read into the computer from punched cards.

You are asked to:
- (1) Build the tables from cards into storage.
- (2) Use the tables to determine the amount to withhold for individual employees.
- (3) Print the withholding tax for each employee.

Part I

Input: (24 cards)

1-6	NET-INC
7-12	EXCESS
13-18	B-TAX
19-20	PCENT

Output:
(Detail line)

NAME
SOCSEC
GR-PAY-Y
W-TAX-M
GR-PAY-M
W-TAX-M

(Heading line)

EMPLOYEE
SOCIAL SECURITY NO.
YEARLY SALARY
YEARLY TAX
MONTHLY SALARY
MONTHLY TAX

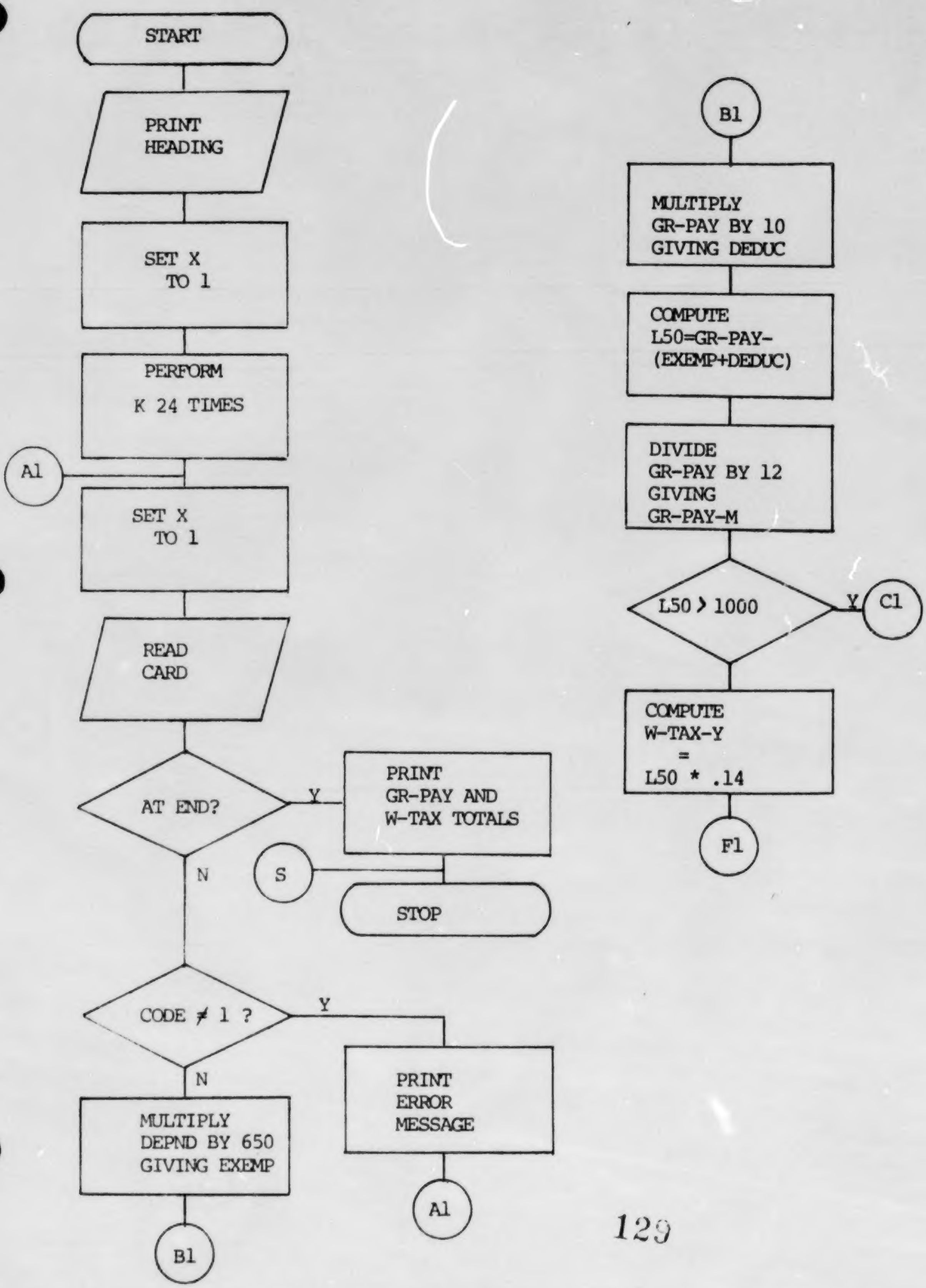
Part II

Input:

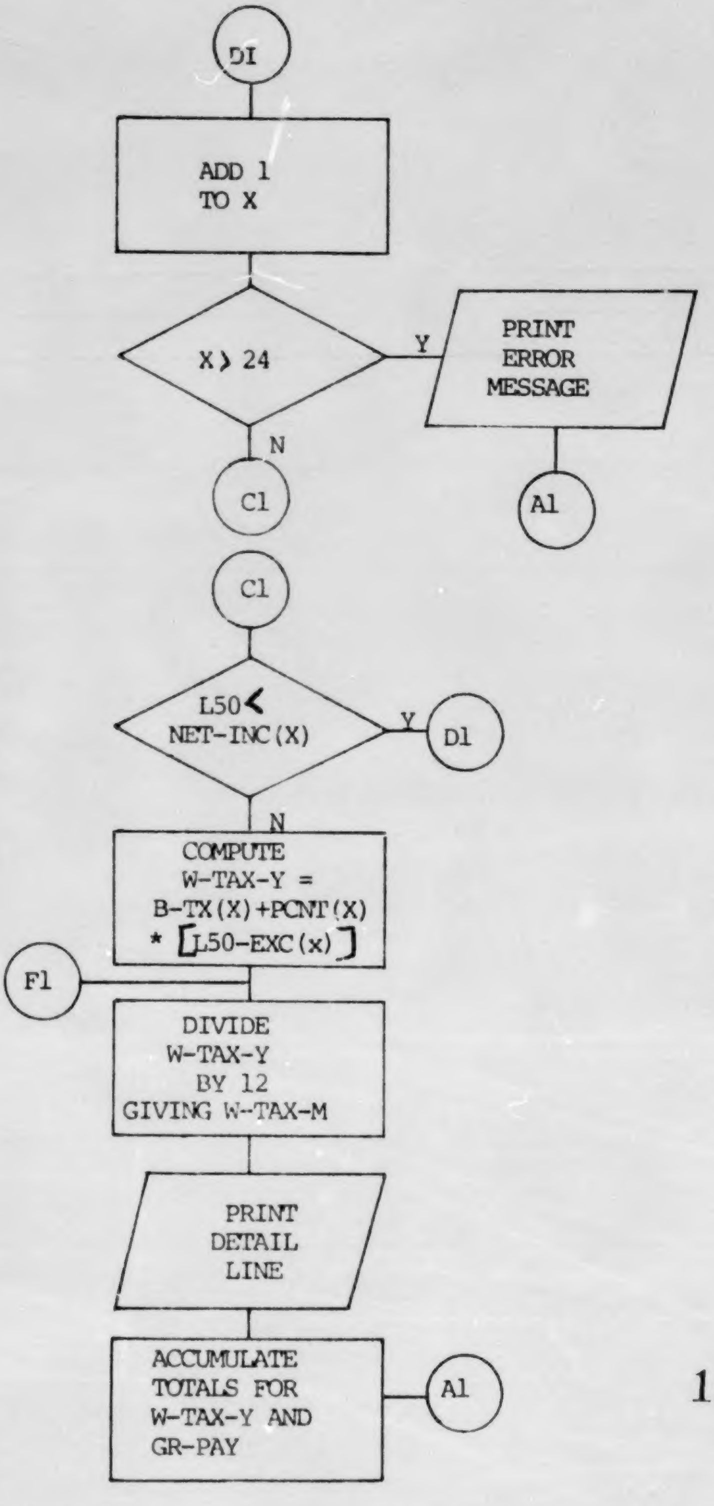
1-20	NAME
21-29	SOC SEC
30-35	GR-PAY
36-37	DEPND
80	CODE-1

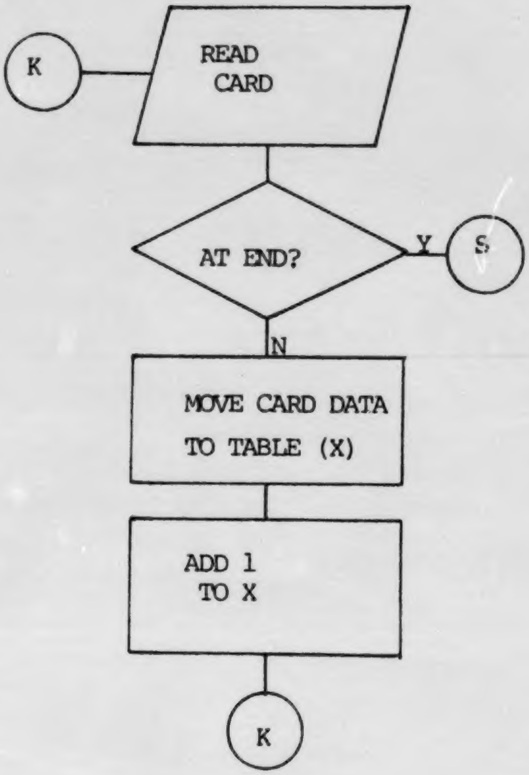
Part I Input cards should be in descending order on Col. 1-6.

EVALUATION 03/06


START
PRINT HEADING
SET X TO 1
PERFORM K 24 TIMES
A1
SET X TO 1
READ CARD
AT END?
Y
PRINT GR-PAY AND W-TAX TOTALS
S
STOP
N
CODE ≠ 1 ?
Y
N
MULTIPLY DEPND BY 650 GIVING EXEMP
B1
PRINT ERROR MESSAGE
A1
B1
MULTIPLY GR-PAY BY 10 GIVING DEDUC
COMPUTE L50=GR-PAY-(EXEMP+DEDUC)
DIVIDE GR-PAY BY 12 GIVING GR-PAY-M
L50 > 1000
Y
C1
COMPUTE W-TAX-Y = L50 * .14
F1

EVALUATION 03/06


DI
ADD 1
TO X
X > 24
Y
PRINT
ERROR
MESSAGE
N
C1
A1
C1
L50 <
NET-INC(X)
Y
D1
N
COMPUTE
W-TAX-Y =
B-TX(X)+PCNT(X)
* [L50-EXC(x)]
F1
DIVIDE
W-TAX-Y
BY 12
GIVING W-TAX-M
PRINT
DETAIL
LINE
ACCUMULATE
TOTALS FOR
W-TAX-Y AND
GR-PAY
A1


K
READ
CARD
AT END?
Y
S
N
MOVE CARD DATA
TO TABLE (X)
ADD 1
TO X
K

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 01: Code and execute simple programs in FORTRAN.

PERFORMANCE GUIDE:

1. Develop and write data names and data descriptions.
2. Finalize flowcharts and other documentation.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job control codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Read lessons 1, 2, and 11 in FORTRAN IV Self-Taught. Study the sections in your manufacturer's FORTRAN reference manual on characters, variables, and constants. Complete Exercises 1-5 in Lesson 2 of FORTRAN IV Self-Taught. Have your work checked.

2. Complete Handout 04/01/02. Have your answers checked.

3. Complete the following assignments:

   a. Read Lesson 3 in FORTRAN IV Self-Taught and complete Exercise 1.
   b. Read Lesson 4 in FORTRAN IV Self-Taught and complete Exercises 1 and 2.
   C. Read Lessons 5 and 6 (through page 46) in FORTRAN IV Self-Taught.
   d. Study comparable sections in your manufacturer's FORTRAN reference manual.

4. Prepare and compile programs for the two problem statements in Handout 04/01/04. Use the procedures listed above in Performance Guides.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare the required FORTRAN program and obtain an error-free compile within four computer runs.

HANDOUT 04/01/02

Part I. Indicate whether each of the following items is a legal or an illegal character or operator in standard FORTRAN.

Item	Legal	Illegal
.LT.		
=		
[		
$		
.ET.		
*		
:		
NE.		
+		
2		
)		
"		
Q		
≤		
A		
6		
%		
/		
M		
-		

HANDOUT 04/01/02

ANSWERS

.LT.	Legal
=	Legal
[	Illegal
$	Legal
.ET.	Illegal
*	Legal
:	Illegal
NE.	Illegal
+	Legal
2	Legal
)	Legal
'	Legal
Q	Legal
≤	Illegal
A	Legal
6	Legal
%	Illegal
/	Legal
M	Legal
-	Legal

Part II. Of the following items, which are constants (CON) and which are variables (VAR)? Which are legal (L) and which illegal (I)? REAL or INTEGER (INT)?

	L	I	CON	VAR	REAL	INT
7.1		X				
A365	X			X	X	
-2196219	X		X			X
C9L1K	X			X	X	
W2.9		X				
3,876		X				
4ALF		X				
+446	X		X			X
-A48		X				
SP+S9		X				
FORTRAN		X				
3HMAD		X				
382.E19	X			X	X	
TANK	X			X	X	
P3	X			X	X	

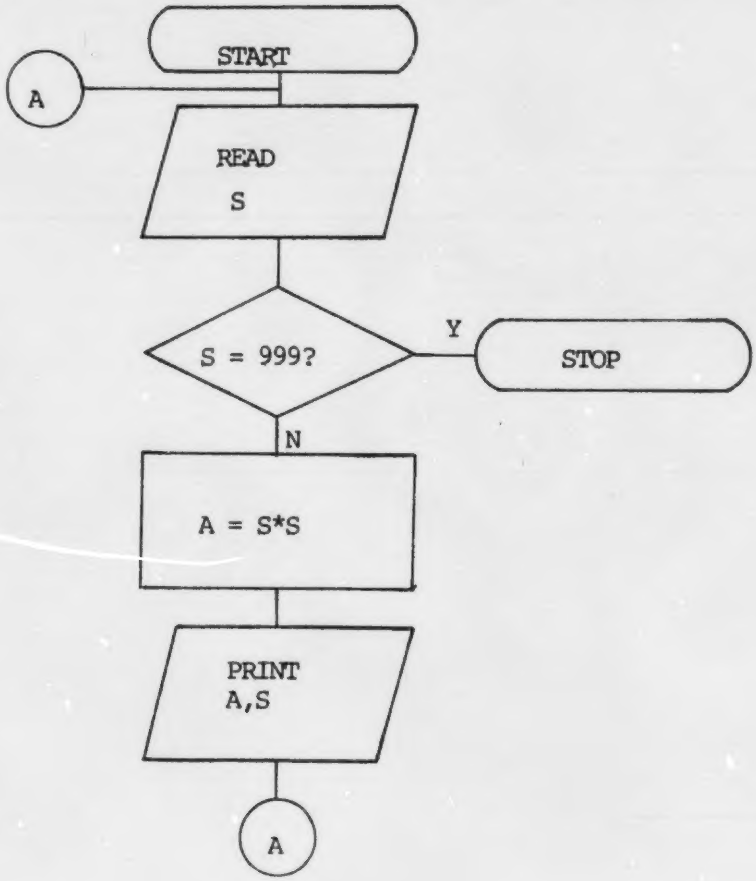
	L	I	CON	VAR	REAL	INT
ELEPHANT		X				
6H123456		X				
.317D+17	X		X		X	
29.1E-51	X		X		X	
499036128587037		X				
TIGER	X			X	X	
+111111111.0	X		X		X	
2A314		X				
(8.5E6,21)		X				
+19.2D56	X			X	X	
5HC+ 21		X				
"KEY-1"		X				
X91115	X			X	X	
78E11	X		X		X	
(4, -22)		X				

HANDOUT 04/01/04

PROBLEM STATEMENT 1

AREA OF A SQUARE

Read the side of a square from a data deck. Calculate the area. Print side and area.


START
A
READ
S
S = 999?
Y
STOP
N
A = S*S
PRINT
A,S
A

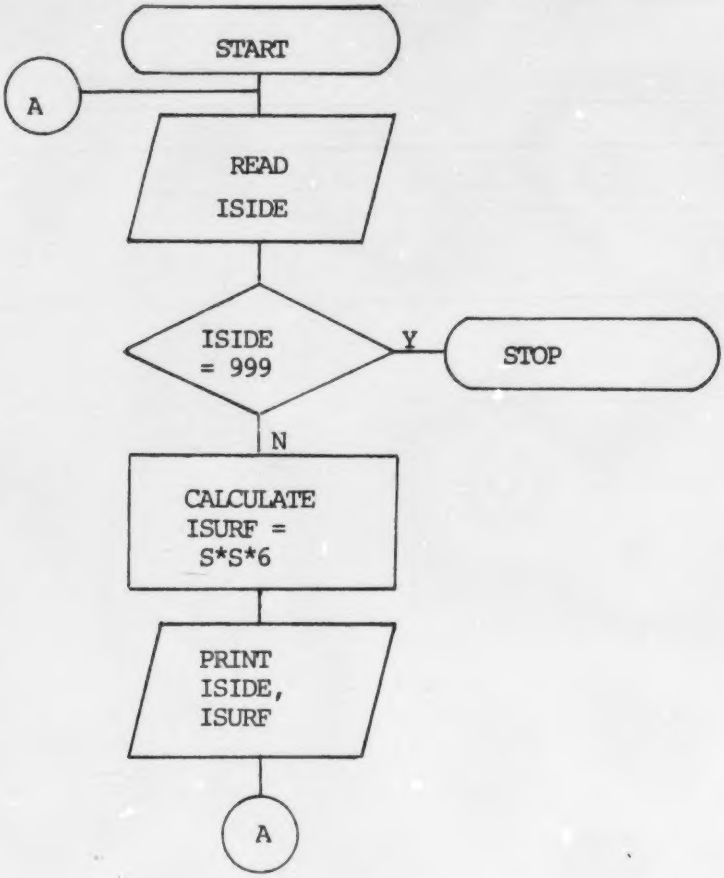
Data Deck

4.0
7.5
16.3
999.

PROBLEM STATEMENT 2

SURFACE OF A CUBE

Given the length of one side of a cube (from a data deck), calculate the total surface area of the cube; print the side length and surface area.


START
A
READ ISIDE
ISIDE = 999
Y
STOP
N
CALCULATE ISURF = S*S*6
PRINT ISIDE, ISURF
A

Data Deck

```
  6
 24
100
999
```

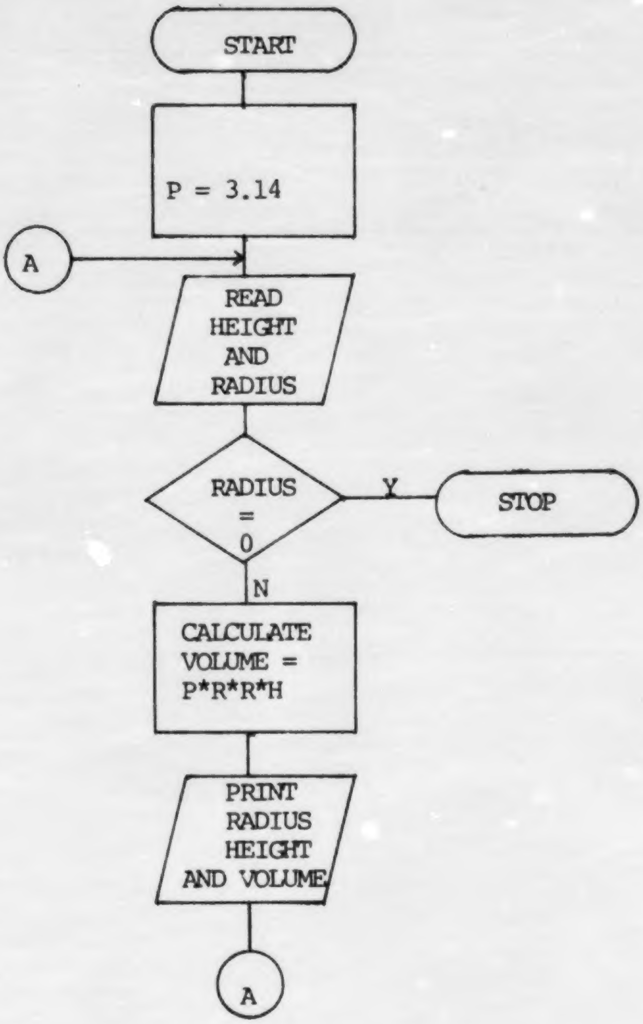
EVALUATION 04/01

VOLUME OF A CYLINDER

To find the volume of a cylinder, use the formula V = (3.14) (R^2) (H) where R is the radius of the base and H is the height of the cylinder.

Use card input.

Print the Base, Height, and Volume in Real Format. Use headings for each column.


START
P = 3.14
A
READ HEIGHT AND RADIUS
RADIUS = 0
Y
STOP
N
CALCULATE VOLUME = P*R*R*H
PRINT RADIUS HEIGHT AND VOLUME
A

Data:

R	H
3.0	6.
4.21	2.85
12.4	82.
	0.

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 02: Use FORTRAN format statements, control statements, and arithmetic operations.

PERFORMANCE GUIDE:

1. Develop and write data names and data descriptions.
2. Prepare flowcharts and other documentation.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job control codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Read Lessons 7, 8, and 9 in FORTRAN IV Self-Taught. Complete Exercises 1-4 for Lesson 7, Exercises 1 and 2 for Lesson 8, and all the exercises for Lesson 9. Have your work checked.

2. Study comparable sections in your manufacturer's FORTRAN reference manual. Pay particular attention to the areas of control statements and library functions.

3. Prepare and compile the programs for the two problem statements in Handout 04/02/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare the required FORTRAN program and obtain an error-free compile within four computer runs.

HANDOUT 04/02/03

For each of the following problem statements, data values are to be read in, the values used in computation, and the results printed. The initial data values are to be printed along with the result for easy reference. For each problem wirte the complete program on FORTRAN coding sheets, enter the codes, and submit the programs for compile and execution.

Problem Statement 1

Read: A, B, C

Print: A, B, C, S, AR

Evaluate:

$$5 = \frac{(A + B + C)}{2} \qquad AR = \sqrt{S\ (S-S)\ (S-B)\ (S-C)}$$

Problem Statement 2

Read: A, B, C

Print: A, B, C, X1, X2

Evaluate:

$$X1 = \frac{-B+\sqrt{B^2 + 4AC}}{2A} \qquad X2 = \frac{-B-\sqrt{B^2+4AC}}{2A}$$

Data for Problems 1 and 2

A	B	C
2	3	4
1	1	1
2	2	8
0	3	6

EVALUATION 04/02

For the following problem statement, data values are to be read in, the values used in computation, and the results printed. The initial data values are to be printed along with the result for easy reference. Write the complete program on FORTRAN coding sheets, enter the codes, and submit the program for compile and execution.

PROBLEM STATEMENT

Read: A, B, C, X

Print: A, B, C, X, R

Evaluate:

$$R = \frac{B.C}{12}\left[6X^2\left(1-\frac{X}{A}\right)^2 + B^2\left(1-\frac{X}{A}\right)^4\right]$$

Data for Problem

A	B	C	X
2	3	4	5
6	27	38	64
5	8	0	3
0	1	2	3

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 03: Debug FORTRAN programs containing syntax errors.

PERFORMANCE GUIDE:

1. Review diagnostic messages against data names and descriptions.
2. Check diagnostic messages against manufacturer's FORTRAN reference manual.
3. Correct program coding errors based on diagnostic messages.
4. Submit program for rerun.
5. Debug program further using subsequent diagnostic messages.

LEARNING ACTIVITIES

1. Each computer manufacturer has its own FORTRAN compiler which produces diagnostic messages as output. Get your manufacturer's FORTRAN reference manual and review the diagnostic messages listed in it.

2. Attend a group discussion on the diagnostic messages listed in your manufacturer's FORTRAN reference manual. Make a list of the most common FORTRAN syntax errors. Also, answer this question: Does the number of diagnostic messages have any relationship to the number of coding errors? Why or why not?

3. From the FORTRAN programs available in your classroom, get a program that has been run and has diagnostic messages. For each message write a probable coding error. Have your work checked.

4. From the FORTRAN programs available in your classroom, get a program that has been run and has diagnostic messages. Debug the program using the procedures listed in Performance Guide.

EVALUATION

Given FORTRAN program with diagnostic messages, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, correct the syntax errors and obtain an error-free compile within two computer runs.

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 04: Use FORTRAN formats I, E, and F for number specifications.

PERFORMANCE GUIDE:

1. Develop and write data names and data descriptions.
2. Develop or revise flowcharts and other documentation to include I, E, and F formats.
3. Code programs on coding sheets.
4. Checking codes for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job control codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Read lessons 12 and 21 in FORTRAN IV Self-Taught. Complete the exercises at the end of each lesson and have your work checked.

2. Study comparable sections in your manufacturer's FORTRAN reference manual.

3. Prepare and compile the program for the problem statement in Handout 04/04/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare the required FORTRAN program and obtain an error-free compile within four computer runs.

HANDOUT 04/04/03

PROBLEM STATEMENT 1

COLUMN DESIGN POINTS

A mechanical engineer wishes to obtain data for plotting a curve of the safe loading of a certain type of column as a function of the slimness of the column. She has selected the following two empirical formulas that give the safe loading in two ranges of the slimness ratio.

$$S = \begin{cases} 17{,}000 - 0.485\, R^2 & \text{FOR } R < 120 \\ \dfrac{18{,}000}{1+\dfrac{R^2}{18{,}000}} & \text{FOR } R \geq 120 \end{cases}$$

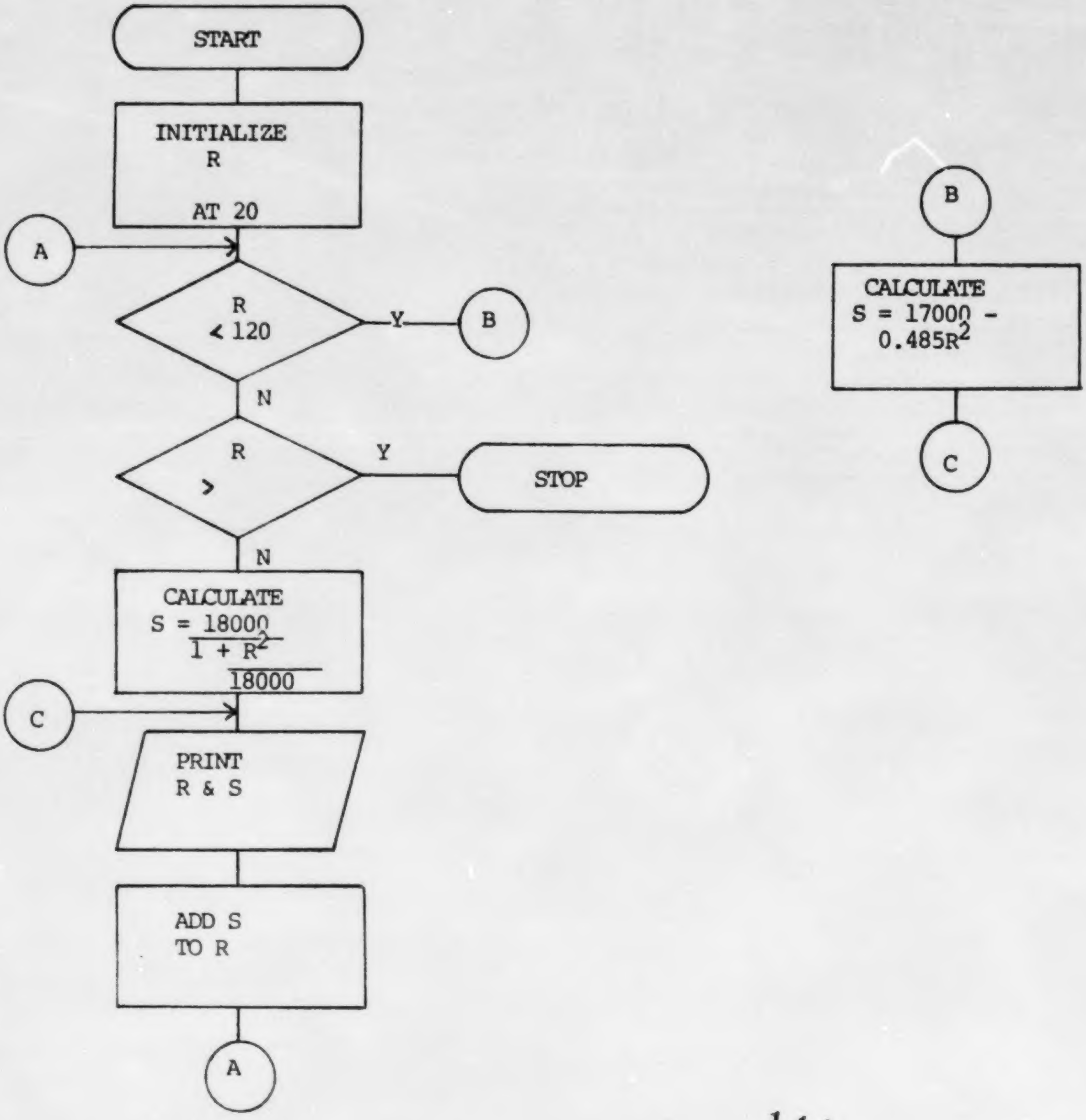
S = Safe load in pounds per square inch
R = Slimness ratio

The safe load is to be calculated for slimness ratios form 20 through 200, in steps of 5. This problem can be done without reading any data cards by writing statements to compute all of the desired values of the slimness ratio. Two IF statements are needed. One will determine which formulas to use, the other will be used to determine when all of the desired values have been used.

Print Output in both E and I formats.

COLUMN DESIGN POINTS - continued

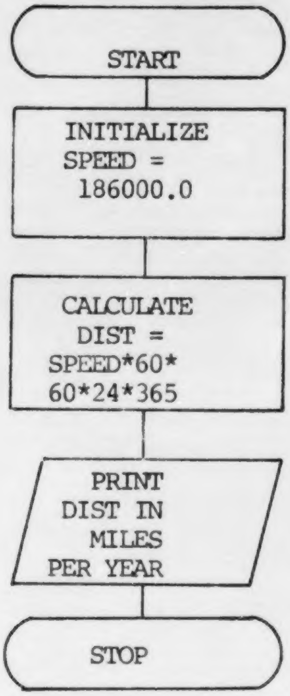
FLOWCHART


START
INITIALIZE
R
AT 20
A
R
< 120
Y
B
N
R
>
Y
STOP
N
CALCULATE
S = 18000
1 + R2
18000
C
PRINT
R & S
ADD S
TO R
A
B
CALCULATE
S = 17000 -
0.485R2
C

EVALUATION 04/04

SPEED OF LIGHT

Light travels at a rate of 186.33 miles per second. Calculate the number of miles it will travel in one normal year. Rewrite, using addition formats for the output. Use the I and E formats as well as the F, and label each one accurately.


START
INITIALIZE
SPEED =
186000.0
CALCULATE
DIST =
SPEED*60*
60*24*365
PRINT
DIST IN
MILES
PER YEAR
STOP

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 05: Use various formats in preparing readable output to FORTRAN problems.

PERFORMANCE GUIDE:

1. Develop and write data names and data descriptions.
2. Develop or revise flowcharts and prepare a printer layout sheet and other documentation.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job control codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Read lessons 13-20 in FORTRAN IV Self-Taught. Complete the first three exercises for each lesson. Have your work checked.

2. Study comparable sections in your manufacturer's FORTRAN reference manual.

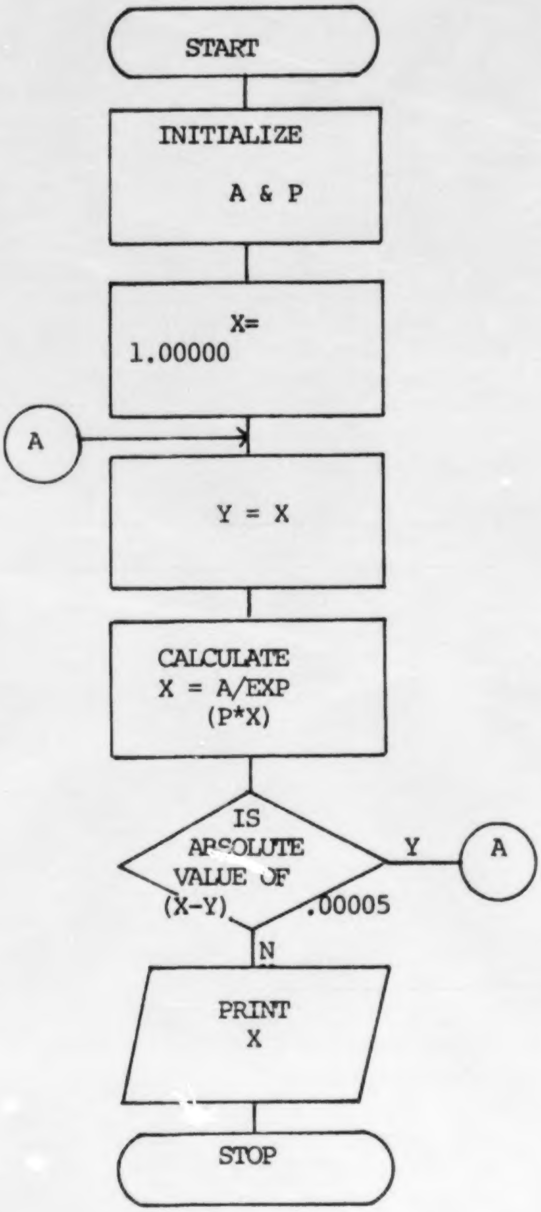
3. Prepare and compile the program for the problem statement in Handout 04/05/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare the required FORTRAN program and obtain an error-free compile within five computer runs.

TRANSCENDENTAL EQUATION

FLOWCHART


START
INITIALIZE
A & P
X=
1.00000
A
Y = X
CALCULATE
X = A/EXP
(P*X)
IS
ABSOLUTE
VALUE OF
(X-Y)
.00005
Y
A
N
PRINT
X
STOP

HANDOUT 04/05/03

PROBLEM 1

DIESEL ENGINE EFFICIENCY

One form of the equation of the theoretical efficiency of a diesel engine is:

$$EFF = 1 - \frac{1}{RY - 1} \quad \frac{(V)^{Y} - 1}{Y\ (V - 1)}$$

Where R = compression ratio
Y = ratio of specific heats
V = load ratio
EFF = efficiency

An engineer wishes to study the variation of efficiency with changes in the compression ratio (R) and the ratio of specific heats (Y) for a fixed value of the load ratio (V). He wishes to input a starting value for R, an increment for R, and the desired number of values of R to be used. The same scheme is used for Y. The efficiency is to be calculated for a fixed value of the load ratio and all combinations of R and Y.

Below is the READ and FORMAT for this problem:

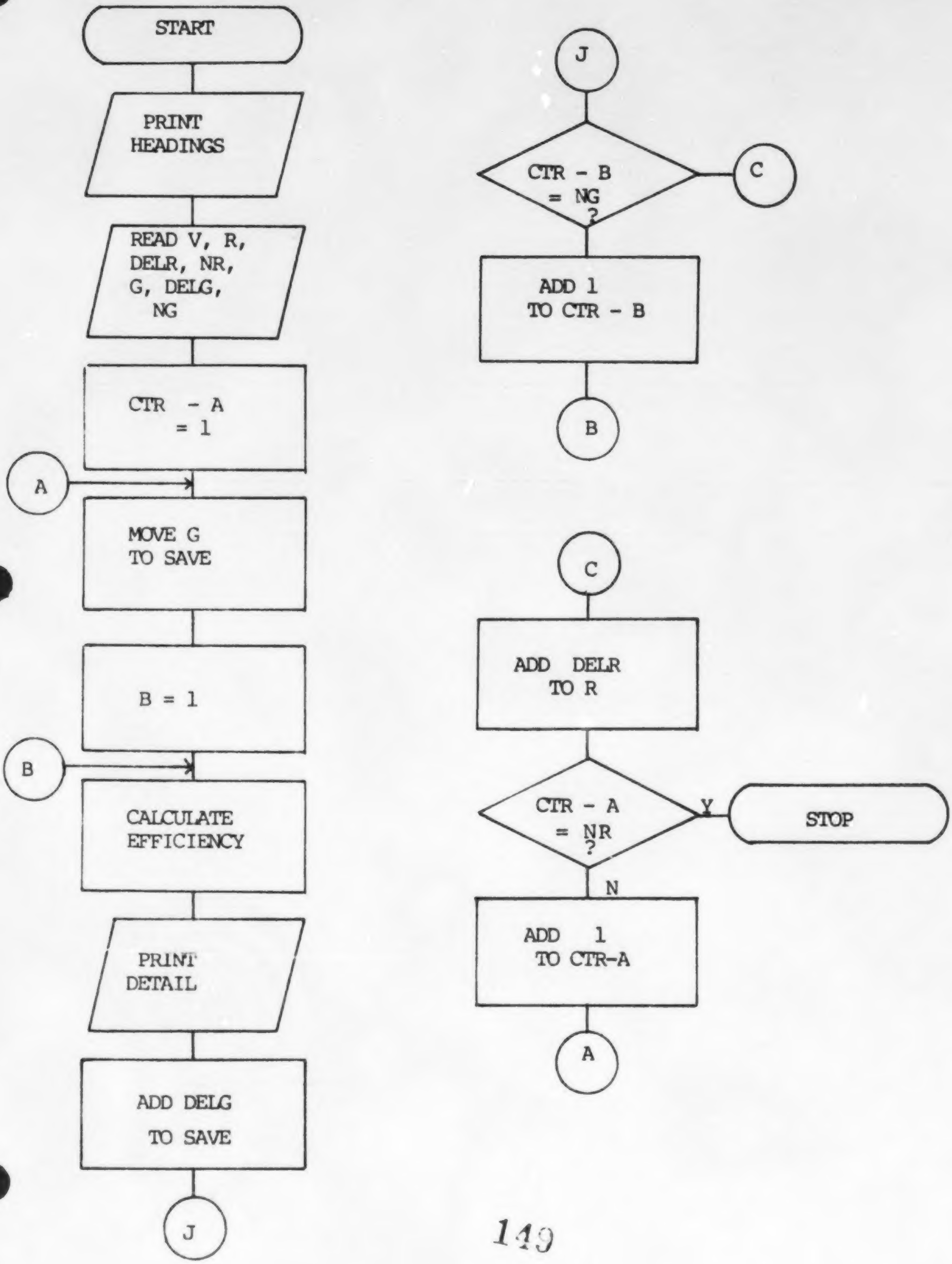
```
READ 30, V, R, DELR, NR, G, DELG, NG
30 FORMAT (3F 10.0, I10, 2F10.0, I10)
```

V = load ratio
R = first compression ratio
DELR = increment in compression ratio
NR = Number of compression ratios
G = first specific heat ratio (Y)
DELG - increment in specific heat ratio
NG = number of specific heat ratios

Set up this problem using a next of two DO loops. The other loop controls the value of R and the enter loop controls the value of Y. A problem of this sort, in which a basic computation is carried out for many combinations of values of the independent variable, is called a parameter study. For output, you are to print the load ratio, compression ratio, heat ratio, and efficiency.

FLOWCHART

04/05/03


START
PRINT HEADINGS
READ V, R, DELR, NR, G, DELG, NG
CTR - A = 1
A
MOVE G TO SAVE
B = 1
B
CALCULATE EFFICIENCY
PRINT DETAIL
ADD DELG TO SAVE
J
J
CTR - B = NG ?
C
ADD 1 TO CTR - B
B
C
ADD DELR TO R
CTR - A = NR ?
Y
STOP
N
ADD 1 TO CTR-A
A

EVALUATION 04/05

TRANSCENDENTAL EQUATION

(Solution of Implicit Formulas by Iterative Methods.)

TRANSCENDENTAL EQUATION

$$x = aeP^{x}$$

The transcendental equation arises frequently in optical absorption problems and in electrical and unclear engineering. Here "a" and "p" are known constants. Field "a" is raised to the exponential function of "p" times "x".

The equation can not be solved explicitly for "x". The following iteration procedure should be used to solve for "x":

1. Make a best guess at "x".
2. Use this in the right hand side of the equation to obtain a new value for "x".
3. Designate this new value of "x" as the next guess.
4. Repeat steps (2) and (3) until two successive guesses either agree or differ by an amount less than or equal to an allowable error.

Best guess for "x" - 1.00000
a = 0.20000
p = 0.50000
allowable error = .00005

STUDENT COMPETENCY SHEET

MODULE 04:	FORTRAN Programming
COMPETENCY 06:	Provide for ease of FORTRAN program maintenance by using various documentation techniques.

PERFORMANCE GUIDE:

1. Develop and write data names and data descriptions.
2. Develop or revise flowcharts and prepare documentation using the techniques of commenting, representing data items with self-explanatory data names, and modular programming.
3. Code program on coding sheets.
4. Check coding for omissions and inconsistencies.
5. Enter program codes in preparation for compile.
6. Proofread codes entered.
7. Enter job control codes in preparation for compile.
8. Submit program for run.
9. Debug program using diagnostics.
10. Test program using test data.

LEARNING ACTIVITIES

1. Read lesson 33 in FORTRAN IV Self-Taught. Complete the exercises at the end of the lesson and have your work checked.

2. Study comparable sections in your manufacturer's FORTRAN reference manual.

3. Prepare and compile the program for the problem statement in Handout 04/06/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare the required FORTRAN program and obtain an error-free compile within five computer runs.

HANDOUT 04/06/03

PROBLEM STATEMENT 1

SUBPROGRAMS - ELECTRIC BILL

PROBLEM : WRITE A FORTRAN MAINLINE PROGRAM WHICH CALLS A FUNCTION SUBPROGRAM AND A SUBROUTINE SUBPROGRAM TO COMPUTE CUSTOMER'S MONTHLY ELECTRIC BILL.

INPUT : DATA CARDS TO BE SUPPLIED. FORMAT OF EACH CARD IS AS FOLLOWS:

CARD COLUMN	DATA FIELD
1	RATE SCHEDULE CODE (1=PRIVATE RESIDENCE, 2=COMMERCIAL ENTERPRISE, OTHER = ERROR).
3-7	CUSTOMER ACCOUNT NUMBER.
8-13	PREVIOUS MONTH'S METER READING.
14-19	PRESENT MONTH'S METER READING.

PROCESSING:

1. WRITE A FUNCTION SUBPROGRAM TO CALCULATE THE BILL OF A PRIVATE RESIDENCE CUSTOMER.
2. WRITE A SUBROUTINE SUBPROGRAM TO CALCULATE THE BILL OF A COMMERCIAL ENTERPRISE.
3. WRITE A FORTRAN MAINLINE PROGRAM TO READ DATA CARDS, CALL THE PROPER SUBPROGRAM, AND PRINT THE CALCULATED BILL.
4. PERFORM EDIT ON RATE SCHEDULE CODE.
5. MAKE THE PROGRAM REPETITIVE.

* RATE CHARGED IS ACCORDING TO FOLLOWING RATE TABLES:

RATE CODE	RANGE	RATE CHARGED
1 (PRIVATE RESIDENCE)	50 KWH OR LESS	$3.00 (MINIMUM BILL)
	51 - 150 KWH	$3.00 PLUS $.05 PER KWH OVER 50
	151 - 250 KWH	$8.00 PLUS $.045 PER KWH OVER 150
	251 - 350 KWH	$12.00 PLUS $ .04 PER KWH OVER 250
	OVER 350 KWH	$16.50 PLUS $.035 PER KWH OVER 350
2 (COMMERCIAL ENTERPRISE)	1000 KWH OR LESS	$35.00 (MINIMUM BILL)
	1001 - 5000 KWH	$35.00 PLUS $.02 PER KWH OVER 1000
	5001 - 20000 KWH	$135.00 PLUS $.015 PER KWH OVER 5000
	OVER 20000 KWH	$360.00 PLUS $.01 PER KWH OVER 20000

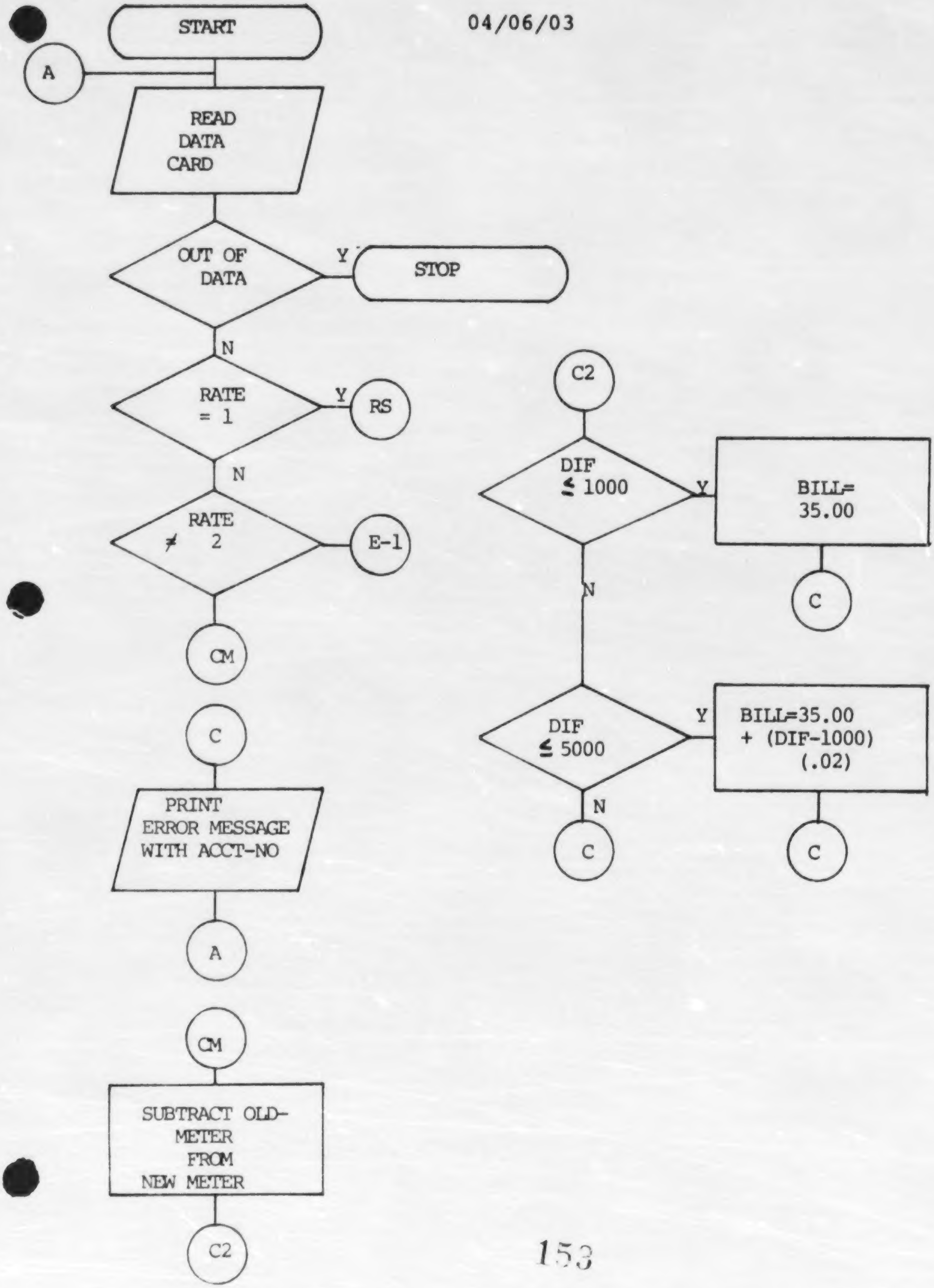
OUTPUT :

1. CUSTOMER BILLING REPORT ON NEW PAGE WITH APPROPRIATE COLUMN HDG.
2. DETAIL LINE (DOUBLESPACED) FOR EACH CUSTOMER CONTAINING RATE CODE, CUSTOMER ACCOUNT NUMBER, PREVIOUS MONTH'S METER READING, PRESENT MONTH'S METER READING, TOTAL UNITS (KILOWATTS USED), AND AMOUNT OF BILL.
3. FLAG ANY ACCOUNT NUMBER IN WHICH ERROR WAS DETECTED IN EDIT, AND DO NOT COMPUTE BILL.

ELECTRIC BILL - continued

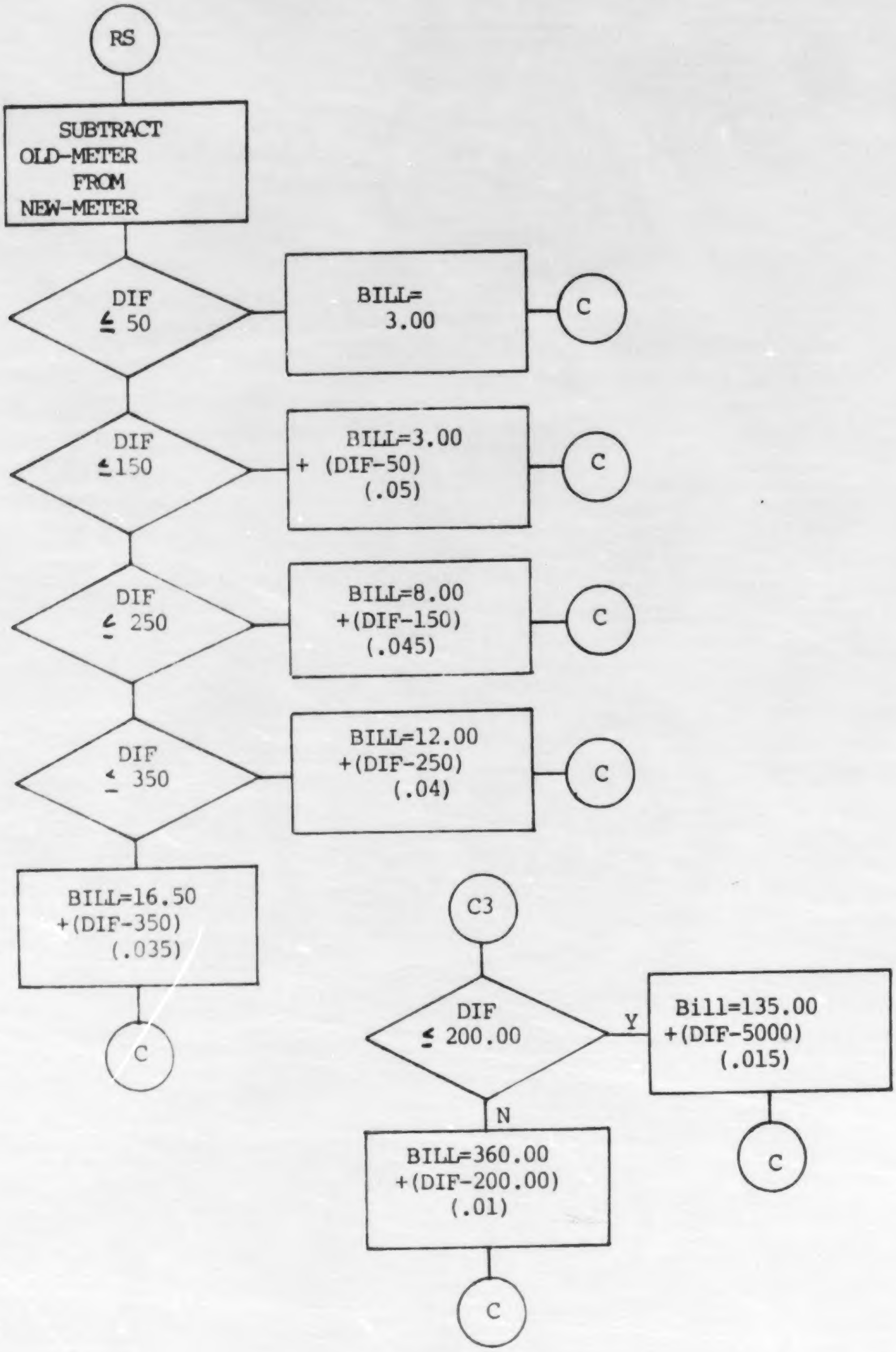
FLOWCHART

04/06/03


START
A
READ DATA CARD
OUT OF DATA
Y
STOP
N
RATE = 1
Y
RS
N
RATE ≠ 2
E-1
CM
C
PRINT ERROR MESSAGE WITH ACCT-NO
A
CM
SUBTRACT OLD-METER FROM NEW METER
C2
C2
DIF ≤ 1000
Y
BILL= 35.00
C
N
DIF ≤ 5000
Y
BILL=35.00 + (DIF-1000) (.02)
N
C
C

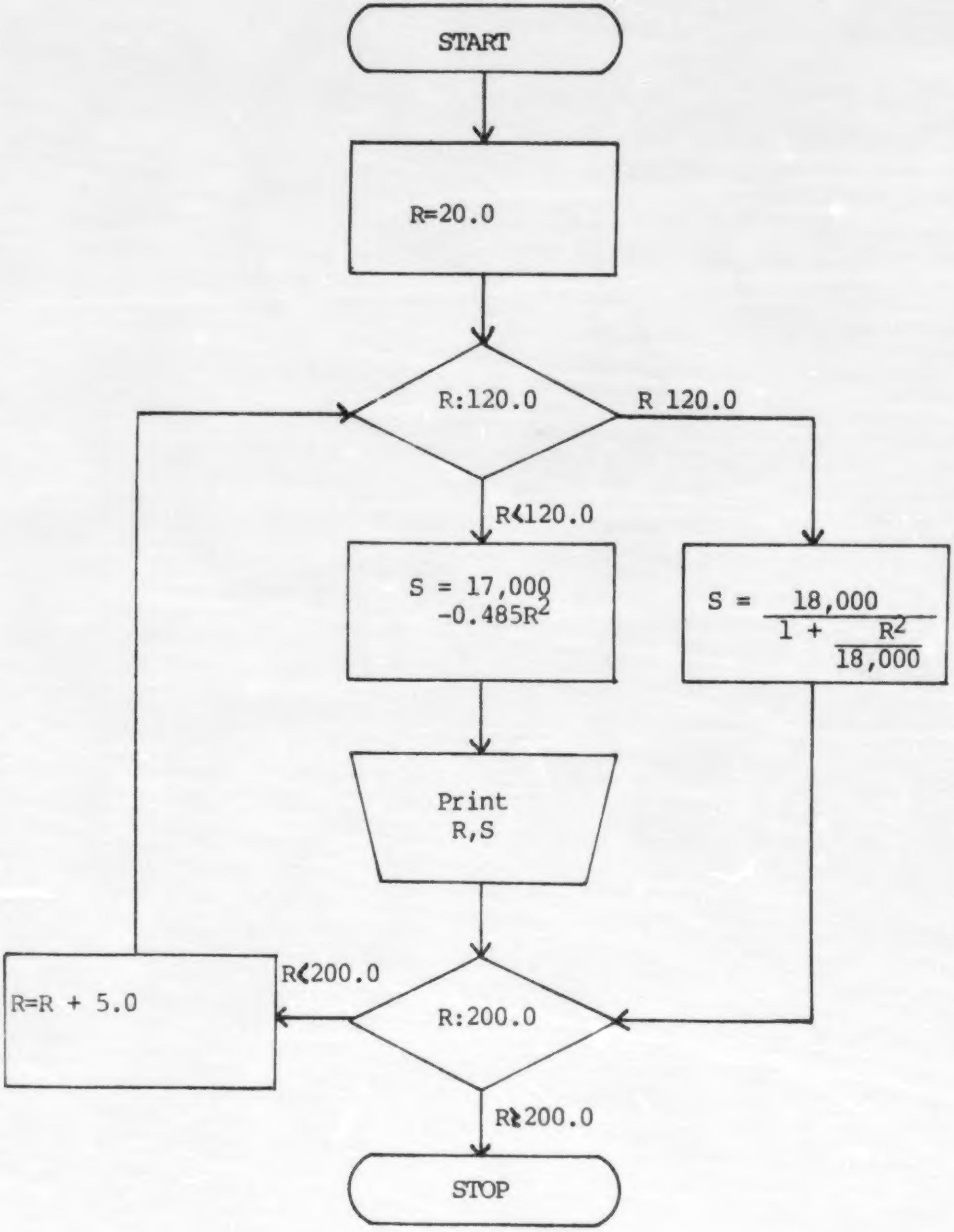
ELECTRIC BILL FLOWCHART

04/06/03


RS
SUBTRACT OLD-METER FROM NEW-METER
DIF ≤ 50
BILL= 3.00
C
DIF ≤150
BILL=3.00 + (DIF-50) (.05)
C
DIF ≤ 250
BILL=8.00 +(DIF-150) (.045)
C
DIF ≤ 350
BILL=12.00 +(DIF-250) (.04)
C
BILL=16.50 +(DIF-350) (.035)
C
C3
DIF ≤ 200.00
Y
Bill=135.00 +(DIF-5000) (.015)
C
N
BILL=360.00 +(DIF-200.00) (.01)
C

EVALUATION 04/06

Code the following problem for values of R from 20 through 200 and execute.


START
R=20.0
R:120.0
R 120.0
R<120.0
$S = 17,000 -0.485R^2$
$S = \frac{18,000}{1 + \frac{R^2}{18,000}}$
Print R,S
R<200.0
R=R + 5.0
R:200.0
R≥200.0
STOP

STUDENT COMPETENCY SHEET

MODULE 04: FORTRAN Programming

COMPETENCY 07: Use the logical statements, the TYPE statements, and format statements of FORTRAN.

PERFORMANCE GUIDE:

1. Develop and write data names and descriptions.
2. Develop or revise flowcharts and other documentation.
3. Prepare test data.
4. Code program on coding sheets.
5. Check coding for omissions and inconsistencies.
6. Enter program codes in preparation for compile.
7. Proofread codes entered.
8. Enter job control codes in preparation for compile.
9. Submit program for run.
10. Debug program using diagnostics.
11. Test program using test data.

LEARNING ACTIVITIES

1. Read lessons 22-24 (through page 218) in *FORTRAN IV Self-Taught*. Complete the first two exercises at the end of each lesson. Have your work checked.

2. Study comparable sections in your manufacturer's FORTRAN reference manual.

3. Prepare and compile the program for the problem statement in Handout 04/07/03. Use the procedures listed above in Performance Guide.

EVALUATION

Given problem statement, flowchart, job control language, test data, manufacturer's FORTRAN reference manual, data entry equipment, and coding sheets, prepare FORTRAN program and obtain an error-free compile within five computer runs.

HANDOUT 04/07/03

PROBLEM STATEMENT 1

LINEAR REGRESSION

THIS PROBLEM REQUIRES THE STUDENT TO WRITE A PROGRAM THAT WILL FIT LINEAR REGRESSION LINES TO SUCCESSIVE SETS OF LATITUDE-TEMPERATURE DATA.

INPUT : DATA ARE PUNCHED AS INDICATED BELOW.

EACH SET OF DATA WILL BE PRECEEDED BY A HEADER CARD CONTAINING A "1" IN COLUMN 1 AND A DATE, IN THE FORM MM-DD-YY IN COLUMNS 5-12, AND THE NUMBER OF RECORDS IN EACH SET IN COLUMNS 15-16. THE LAST DATA SET WILL BE FOLLOWED BY A CARD CONTAINING A 9 IN COLUMN 1. INDIVIDUAL DATA RECORDS WILL HAVE THE CITY PUNCHED IN CC 4-19, LATITUDE PUNCHED IN CC 22-25 (xx.x), AND TEMPERATURE IN CC 30-31 (xx)

PROCESSING:

WRITE A PROGRAM TO FIT LINEAR REGRESSION LINES TO SUCCESSIVE SETS OF LATITUDE-TEMPERATURE DATA. FOR EACH SET OF DATA, COMPUTE THE PARAMETERS A AND B IN THE EQUATION TEMP = A + B(LAT) WHERE: (E IS USED FOR SUMMATION SIGN)

```
         E (LAT(I))**2 E(TEMP(I))- E(LAT(I)) E(LAT(I)TEMP(I))
   A = -----------------------------------------------------
             N  E(LAT(I))**2    -   (E(LAT(I)))**2

         N E(LAT(I)TEMP(I))   -  E(LAT(I))  E(TEMP(I))
   B = -----------------------------------------------------
            N    E(LAT(I))**2    -   (E(LAT(I)))**2
```

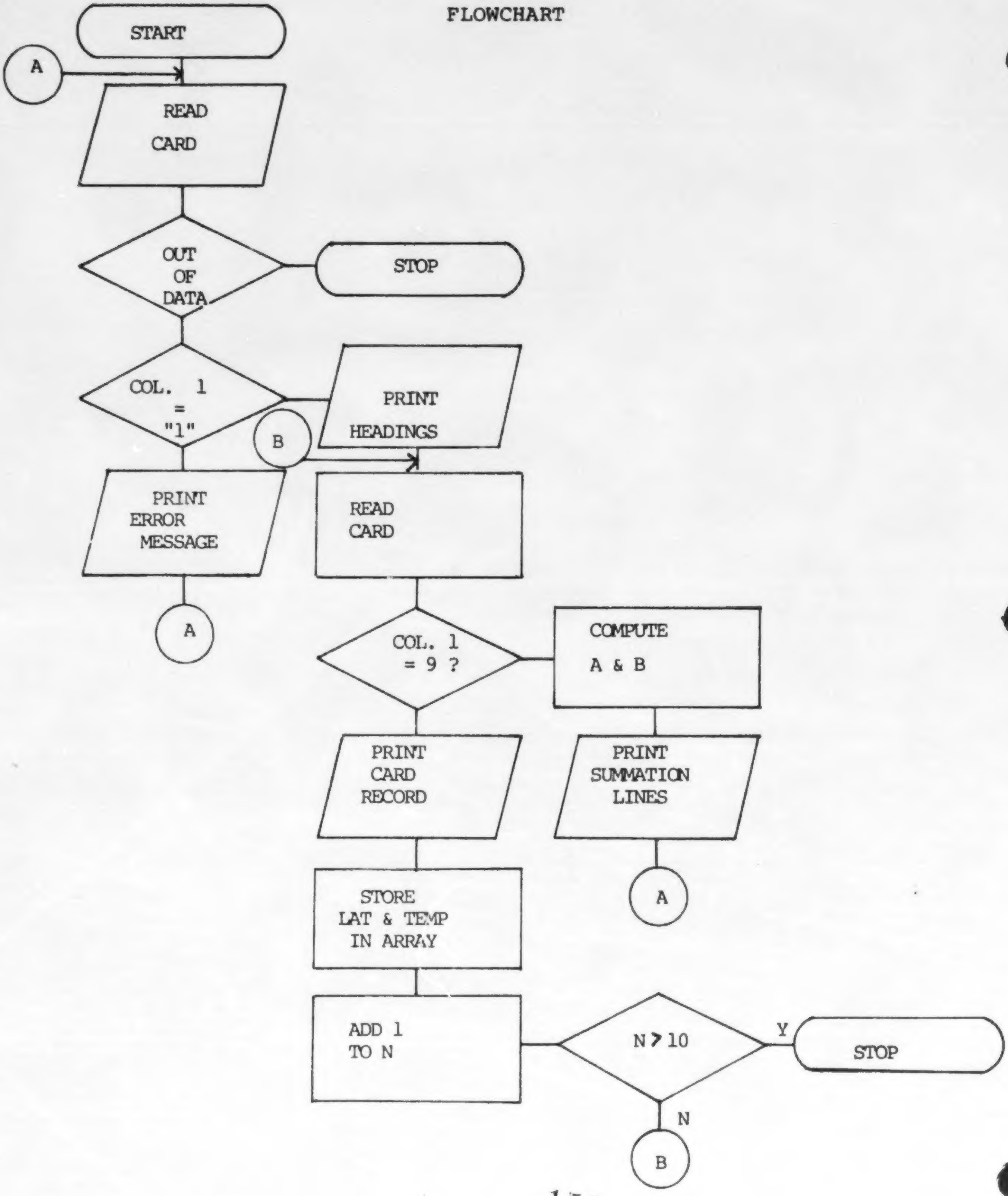
OUTPUT : FOR EACH SET OF DATA, PRINT A HEADER STARTING ON A NEW PAGE, FOLLOWED BY THE RECORDS IN THE DATA SET, FOLLOWED BY 2 LINES WHICH READS:

```
A  =  xx.xx  B = xx.xx  SUMLAT =  xxxx.xx
SUMTEMP = xxxx.   SUMLATSQ = xxxxxx.xx   SQSUMLAT = xxxxxx.xx
```

INCLUDE IN YOUR REPORT A PLOT OF THE DATA AND THE COMPUTED REGRESSION LINE FOR THE SECOND DATA SET.

LINEAR REGRESSION - continued

FLOWCHART


START
A
READ
CARD
OUT
OF
DATA
STOP
COL. 1
=
"1"
PRINT
HEADINGS
B
PRINT
ERROR
MESSAGE
READ
CARD
A
COL. 1
= 9 ?
COMPUTE
A & B
PRINT
CARD
RECORD
PRINT
SUMMATION
LINES
A
STORE
LAT & TEMP
IN ARRAY
ADD 1
TO N
N > 10
Y
STOP
N
B

EVALUATION 04/07

POPULATION DENSITY

THE DICTATOR, BEN E. FICENCE, OF LOWER & HIGHER PURGATORY DESIRES SOME FORMATION ABOUT POPULATION DENSITIES IN HIS COUNTRY, THE LATEST CENSUS, 1971, ABOUT CENTRAL CITIES (CC) WITH A POPULATION OF OVER 1 MILLION UMAMIABLE RESIDENTS, AND THEIR STANDARD METROPOLITAN STATISTICAL AREA (SMSA) WHICH INCLUDES THE CENTRAL CITIES.. DATA GATHERED ALSO INCLUDED THE AREA IN CUBITS OF THE CC AND SMSA LOCATIONS.

A. BEN E. FICENCE DESIRES THE POPULATION DENSITY FOR EACH CC AND SMSA RING (EXCLUDING CC) AND THE AREA OF EACH CC AS A PERCENT OF THE AREA OF THE SMSA.

B. CENSUS DATA IS AS BELOW --6 RECORDS

CC	FIELD
1-16	CITY NAME
17-23	CITY POPULATION
24-30	SMSA POP.
31-33	AREA OF CC
34-36	AREA OF SMSA

OUTPUT SHOULD APPEAR AS BELOW.

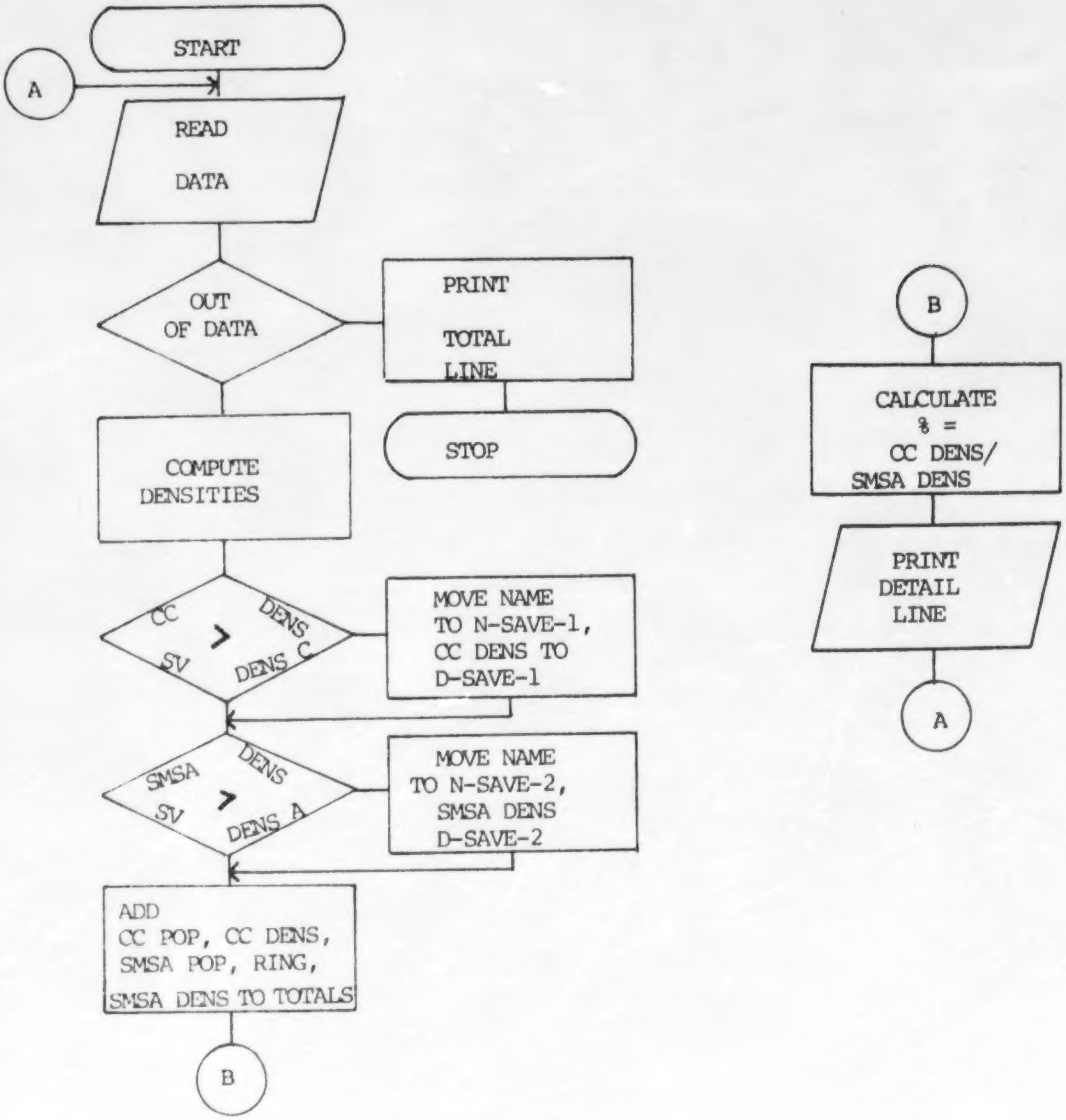
```
 CITY   CC    SMSA   CC    SMSA   CC    SMSA   SMSA RING   CC % OF
NAME   POP     POP  AREA   AREA  DEN    DEN    DENSITY     SMSA AREA
 xx     xx     xx     xx     xx    xx   xx     xx    xx     xxx
```

TOTALS SHOULD BE PRINTED FOR THE COLS.S OF CC POP. SMSA POP AND AREAS

****** FIND AND IDENTIFY THE CC AND SMSA WITH THE HIGHEST POPULATION DENSITY IN LOWER & HIGHER PURGATORY.

POPULATION DENSITY - continued

FLOWCHART


START
A
READ
DATA
OUT
OF DATA
PRINT
TOTAL
LINE
STOP
COMPUTE
DENSITIES
CC
DENS
>
SV
DENS C
MOVE NAME
TO N-SAVE-1,
CC DENS TO
D-SAVE-1
SMSA
DENS
>
SV
DENS A
MOVE NAME
TO N-SAVE-2,
SMSA DENS
D-SAVE-2
ADD
CC POP, CC DENS,
SMSA POP, RING,
SMSA DENS TO TOTALS
B
B
CALCULATE
% =
CC DENS/
SMSA DENS
PRINT
DETAIL
LINE
A